# Data-Driven Rehabilitation Development for Stroke Patients Using Machine Learning Techniques

by

**Marcus C. Allen**

B.S. in Mechanical Engineering, University of Rhode Island, 2014

Submitted to the Graduate Faculty of the

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2020

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

**Marcus C. Allen**

It was defended on

November 8, 2019

and approved by

Daniel G. Cole, Ph.D., Associate Professor
Department of Mechanical Engineering and Materials Science

Amit Sethi, Ph.D., Assistant Professor
Department of Occupational Therapy

Nitin Sharma, Ph.D., Associate Professor
Department of Biomedical Engineering
University of North Carolina at Chapel Hill

Dissertation Director: William W. Clark, Ph.D., Professor
Department of Mechanical Engineering and Materials Science

**Data-Driven Rehabilitation Development for Stroke Patients Using Machine Learning Techniques**

Marcus C. Allen, PhD

University of Pittsburgh, 2020

A portable motion capture system that can guide a stroke patient's rehabilitation session will help them regain motion in their arm quicker and save money on medical bills. The goal of this dissertation was to investigate stroke patient motions to lay the groundwork for such a system. Specifically, the dissertation focused on developing an automated system for stroke diagnostics and task-oriented rehabilitation. This form of telerehabilitation is new to the field since previous systems do not create skill-based instructions for the patient.

The system created in this dissertation was demonstrated to classify a patient's stroke severity using supervised machine learning techniques. It was shown how this information can be used to create a personalized rehabilitation protocol for that patient. For this study, inertial measurement units were used to collect joint kinematics and kinetics during reaching tasks for 10 healthy and 17 stroke subjects, and from that data machine learning features were defined for a support vector machine classification algorithm. These parameters were validated against a gold standard optical tracking motion capture system (Optitrack, NaturalPoint, Inc.) where the joint results were calculated using the Motion Monitor Biomechanics engine. The kinematic and kinetic features extracted from these patients for the machine learning model also served a dual purpose by creating a hierarchal rehabilitation session by categorizing reaching tasks by difficulty. The features used to determine a patient's task hierarchy for rehabilitation were determined based on the support vector machine algorithm that classified stroke severity. Task

rankings were developed for stroke individual stroke subjects as well as for classes of stroke subjects. The rankings were unique for classes and for individuals, but their general orders coincided with intuitive task difficulty.

## Table of Contents

## List of Tables

## List of Figures

## Acknowledgements

I would like to start my dissertation by acknowledging my advisor Dr. William Clark. I could not have made it here without his unwavering support. Through his guidance, I was able to grow and learn difficult concepts in a short amount of time. I also would like to thank my committee members (Drs. Nitin Sharma, Daniel Cole, and Amit Sethi), for allowing me to work on their projects and for their support. Special thanks, goes out to Dr. Sylvanus Wosu for giving me a warm introduction during my visit to the University. Furthermore, I would like to acknowledge Dr. David Kristo and his family for allowing me to stay with them during my last two months in Pittsburgh. Their act of kindness made my transition from student to a working professional that much easier. Lastly, I want to acknowledge my family and friends for their steadfast belief in my abilities and for their support throughout graduate school. As I transition to my professional career as a Systems Engineer at Raytheon, I will be forever grateful for the time everyone has invested in my professional and personal growth. It would not have been possible without their help.

# 1.0 Introduction

As of 2018, there were 795,000 new or reoccurring cases of stroke in the United States expected each year (Benjamin E.J., 2018). A maximum of 85% of these stroke cases result in an individual having upper extremity paresis poststroke (Kwakkel G., 2007; Sethi A., 2017). Unfortunately, this is a large responsibility for our healthcare system with the total direct and indirect cost being $65 billion as of 2008. The direct cost includes hospitalization, nursing home healthcare professionals and drugs. While indirect costs consist of loss of productivity due to morbidity and mortality (Rosamond, 2008). These patients need to start rehabilitation immediately, so they can regain their motor skills and save money on their medical bills. This dissertation focuses on upper extremity stroke patients and provides knowledge on data-driven based rehabilitation.

A challenge with rehabilitation is the high cost (time and money) of direct contact between patients and clinicians, and the difficulty that patients have properly following a rehab routine on their own. Patients currently travel to the clinician to be diagnosed and treated and are given a series of tasks after each session to complete at home to continue their rehabilitation. With the lack of guidance outside of the clinic, patients tend to perform the exercise incorrectly and become unmotivated. The completion of each exercise is qualitative and can provide a patient with a false sense of success.

A user-friendly system that helps the patient regain normal upper extremity functions and avoids the development of irregular motion to compensate will help with the disconnection between clinician visits. In order to create such a system, more information is needed about stroke patient motions to characterize the motions, classify the severity of stroke impairment, and

prescribe rehabilitation treatment that is individually suited for each patient. The research described in this dissertation is intended to provide information in these areas, with the ultimate goal of accelerating the rehabilitation process for stroke survivors who only receive professional feedback during clinician visits.

Several key aspects of an eventual portable and individualized rehabilitation system are proper diagnosis and classification of a patient's stroke severity, prescription of proper individualized therapy, and monitoring of progress. Inertial measurements can assist in each of these areas.

Diagnosis of stroke severity is challenging and requires significant training. Clinicians also face difficulties when evaluating their patients. During an evaluation, clinicians can only measure the range of motion (ROM) of one joint angle at a time with a goniometer. Due to this limitation, clinicians cannot measure movement during dynamic functional tasks, which makes it difficult for them to establish accurate rehabilitation goals. A system that provides the clinician with quick, objective and automated methods of assessment of motor severity after stroke using kinematic and kinetic measurements will help facilitate a productive rehabilitation regime.

Once a therapy has been prescribed, clinicians lack necessary time to work with patients one-on-one in order to deliver high quality repeated task-oriented rehabilitation. With the lack of professional guidance, this can cause the stroke patient to perform the task incorrectly and can lead to reinforcement of negative compensatory motor strategies. A system is needed that can have patients monitor their hand improvement and receive therapist-guided personalized treatment to accelerate and improve functional independence.

This dissertation describes research that explores a wearable motion measurement system and accompanying analysis techniques that can lay the foundation for a system that may provide the benefits needed in stroke rehabilitation.

## 1.1 Dissertation Outline

This dissertation focuses on establishing the underlying information and methods that are needed to ultimately develop a data-driven customizable rehabilitation system. Specifically, there are three aims of the research:

Aim 1: Define a kinematic and kinetic based feature set (including new features and existing from the literature) from which assess and predict stroke severity (Chapter 3.0). Since in this work we are using wearable IMU sensors as the means of measuring patient motions, two sub-aims are created: i) verify the IMU system's accuracy against the gold standard optical tracking system (Chapter 4.0). ii) develop algorithms with which to calculate those features from raw IMU data.

Aim 2: Illustrate how the features defined in Aim 1 can be used to predict stroke severity (Chapter 5.0).

Aim 3: Show how the classification results can be used to create a task difficulty hierarchy for a patient classes (of similar stroke severity) and for individual patients (Chapter 5.0) that may, in turn, be used to prescribe personalized treatment.

The measures of success to ensure that each aim was completed are listed below.

Aim 1 measures of success:

- **Define a set of kinematic and kinetic features to assess and predict stroke severity:** Previous studies have analyzed how a single motion feature correlates with stroke severity, but a larger set of features needs to be considered for predictive analysis (machine learning) to be applied. The aim is complete when kinematic and kinetic based features that address, all joint motions, range of motion, and the relationship between joints are identified and quantified in this dissertation. This is an iterative process that requires revisiting to define new features if the measure of success for Aim 2 is not met.

- **Calculate motion features from the raw data that predict stroke severity:** This aim is complete when all motion features defined in the first measure of success of this aim (kinematic and kinetic based) are calculated from the IMUs.

- **IMU kinematic validation:** To ensure that the IMU's joint angle readings are accurate, the IMU results are compared to an optical tracking system (Optitrack, © 2018 NaturalPoint) which is the gold standard in motion capture technologies. The goal is to have the normalized cross-correlation between the two systems be within 0.8 – 1, where a normalized cross-correlation of 0 is the worst performance and 1 is the best performance.

Aim 2 measures of success:

- **Prediction of stroke severity using motion features:** Using the features defined in Aim 1, feature sets are to be selected for use in machine-learning classification methods. F1 scores of resulting classifications are to be used to quantify the performance of the classification. The aim is complete when the average F1 score over 150 random iterations of training sets is within 0.95) – 1.

Aim 3 measures of success:

- **Automated goal-oriented rehabilitation development:** The goal is to develop a method with which to rank rehabilitation tasks based on the difficulty for an individual stroke subject or a class of stroke subjects. Two measures are considered to complete this aim. The first measure is the ability to quantify motor impairment for each stroke participant. This can be shown through the deviation of motion features from a healthy population as stroke severity increases. Success is defined when a measure is defined that enables ranking of tasks in relation to their difficulty for a stroke subject. The second measure is the ranking of the tasks based on difficulty.

It is important to note that the results presented in this dissertation outline a method that can be used if a large patient population (100+ participants) for study. Due to the limited dataset available in this work (27 subjects), conclusions cannot be made on the exact subset of features selected and the exact task hierarchy for any given class of stroke subjects (explained in detail in Chapter 5.0 and 6.0), but the methods themselves are expected to be scalable. Therefore, the broader impact of this work leans more on the method proposed. Achieving success in the aims will provide strong evidence that a more comprehensive study with more subjects will solidify the method.

## 2.0 Background and Motivation

Current stroke rehabilitation for upper extremity impairment is a cyclic process that consists of three steps. These steps include (1) assessment, to identify and quantify the patient's needs (2) goal setting, to define realistic and attainable goals for improvement, and (3) intervention, to assist in the achievement of the goals (Langhorne P., 2011). The following sections describe the standard methods for each step along with recent developments to provide more quantitative data to expand knowledge on rehabilitation approaches.

### 2.1 Stroke Assessment

Stroke impairments are currently assessed by a clinician using the Fugl-Meyer Upper Extremity (FMUE) assessment (Yang Q., 2017). This test focuses on the ROM of the joint angles and the speed at which a patient performs a task. Patients are asked to perform a total of 33 tasks (The list of tasks can be seen in Appendix A) that involve the patient fully extending or flexing a given joint and grasping household goods such as a soda can. Each task is given a score of 0 for not being able to complete the task, 1 for partial completion and 2 for full completion (Singer B., 2016). The FMUE scale is set such that 66 is the maximum score (healthy patient) and 0 is the minimum score (a patient who is severely limited in completing tasks). Although the FMUE is a strong indicator of stroke severity, the static joint measurements (i.e. ROM) cannot capture the compensatory reaching strategies developed after stroke. Also, Fugl-Meyer and Gladstone (Fugl-Meyer A.R., 1975; Gladstone D.J., 2002) state that the FMUE has difficulty

distinguishing the difference between improvement in motion and improvement in strength. Through dynamic measurements from motion capture technologies, compensatory strategies can be detected, and telerehabilitation (rehabilitation services in remote sites (Peretti A., 2017)) methods can be developed to correct their motion.

Researchers in the biomechanics field have been studying human motion tracking for rehabilitation since the 1980s (Zhou H., 2007). Throughout this time, the size of the motion tracking devices has decreased while the computational power has increased. Researchers have benefitted from highly accurate optical tracking systems such as VICON (VICON Industries, Inc) but are not applicable for rehabilitation at home due to the system's large size, significant cost, and complexity of setup and use. In a search for portable systems, researchers have been using inertial sensors, encoders (potentiometers) and pressure sensors to analyze limb kinematics.

Even though encoders and force plates are small, they still have some significant drawbacks in the motion tracking field. Encoders require some form of a brace to attach them to the body and each encoder can only track one axis of rotation. While pressure sensors are highly accurate, they are currently limited to measuring bending or twisting along the surface to which they are attached and need to be in contact with the ground. In some configurations, the pressure sensors can restrict the motion of the subject. Inertial measurement units (IMUs), on the other hand, are promising for this field due to their low cost, small configuration (no additional attachments and/or ground contact needed), and high performance through sensor fusion of the accelerometers, gyroscopes, and magnetometers. Also, IMUs and other wearable sensors have

already found widespread applications in homes, with examples such as the Wii U system (Nintendo Co., Ltd), Fitbit and smartphones.

Stroke rehabilitation with IMUs has been explored by Willmann et al. (Willmann R.D., 2007), where the patient and clinician received feedback from the system in the form of joint angles in the sagittal, frontal and horizontal body planes. While this is a significant advancement in the field, there still is a lack of quantitative feedback for motion strategies that lead to recovery outside of the clinic. A system that can use historical data to provide a personal path to recovery based on the patient's level of impairment would help avoid stagnant recovery during home rehabilitation sessions. In order to create this system, it would need to be capable of properly predicting a stroke patient's impairment from their motion patterns.

Studies have introduced pattern recognition in motion tracking to identify a patient's disability. B. Najafi et al, have used discrete wavelet transform of a gyroscope signal to identify sit-to-stand transitions in elderly patients (Najafi B., 2002). Mechanical impedance has been correlated with the Unified Parkinson's Disease Rating System to diagnose rigidity in Parkinson patients using a force transducer and a gyroscope (Patrick S.K., 2001). In each study, the output from the system was used in a scalar equation to identify the motion. Identifying stroke motion can be more complex since multiple joint kinematics need to be monitored. One of the objectives of this research is to develop an IMU-based system in which multiple limb movements and joint angles can be measured simultaneously.

Solely using joint kinematics can cause difficulties in distinguishing unique patterns between different motions. Additional parameters are needed to define the sporadic motion in

stroke patients. In the Biomechanics field, kinetic features have been developed to describe how the neuromuscular system effectively organizes the multiple degrees of freedom movement.

Kinetic features can give unique outputs for sporadic behavior that have a high correlation with stroke severity. Hirashima (Hirashima M., 2003) has observed the interaction torque (the sum of the Coriolis torques and torques due to the angular acceleration of other joints) to quantify joint coordination throughout a motion. The remainder of the torque terms were defined as net torque (torque due to angular acceleration of the joint of focus), gravity torque, and muscle torque (calculated as the net torque minus the sum of the interaction and gravity torques). To avoid confusion with the standard definition of net torque (sum of all torque terms), the Hirashima definition of net torque will be called the focal point torque for the remainder of the dissertation (further details of this calculation are in section 4.4). When these torques were analyzed with patients that have damage to their central nervous system through Cerebellar Ataxia, the patient adjusted their muscle torques to account for the change in interaction torque patterns (Bastian, 1996). The results demonstrated in Bastian's study show that the central nervous damage caused by a stroke can be captured in gravity, interaction, focal point, and muscle torques.

As more and more motion features are considered (e.g. kinematic and kinetic information from multiple joints and body segments), a great deal of data is created. With the high volume of information, it is easy for a clinician to overlook motion patterns that predict stroke severity. Data-driven methods such as machine learning have been used to identify patterns in large datasets and predict future outcomes.

Algorithms such as artificial neural networks (ANNs), Naïve Bayesian classifiers, gaussian mixture models and SVMs have been used to recognize daily and sports activities

(Barshan, 2013). In that work, the SVM and ANNs had an identical classification performance by evaluating the sports activities with an accuracy of 99.2%. What makes the SVM advantageous between the two is its robustness to noise in the data training set and the transparency of the predictive model. When patients are affected by stroke, their spatio-temporal irregularities provide additional complexity (or process noise) in the motion characteristics due to jerk (change in acceleration over time) (Osu R., 2011).

By following the machine learning methods in the previous studies, one objective of this dissertation was to develop a machine-learning algorithm to identify patterns in the high dimensional datasets that will be created. With this technology, an initial input dataset (kinematic and kinetic data) will be paired with the desired output known to the user (FMUE score) and will be used as the training set for the machine learning algorithm. When implemented correctly, this process produces a function (or predictive model) that can make appropriate patient assessment decisions with future datasets without the guidance of the clinician.

## 2.2 Goal Setting

Once the patient's stroke severity has been classified, the main goal is to correct their motion over time. Current clinical rehabilitation focuses on repetitive task-oriented training as the gold standard motor learning method for stroke patients (Winstein C.J., 2014). In a clinical setting, the first step involves choosing the appropriate tasks (2. Goal setting). Winstein states that goal setting is broken down into 3 objectives: (1) the task must be challenging enough to require new learning, (2) the training must progress in difficulty and the task must be iteratively adapted to real-world relevance and (3) the task must not lead to rote repetition but repeated attempts to solve

a problem. To enforce the proper task-oriented regime a clinician needs to find a series of tasks that are achievable for a stroke participant.

Clinicians prescribe challenging achievable goals using the FMUE assessment by selecting FMUE elements that they scored a 1 on (partially completed the task). For example, if a stroke patient scored a 1 on the spherical grasp task during the FMUE assessment, they will be asked to perform that task during their rehabilitation session (FMUE tasks will be explained in Chapter 3.0).

## 2.3 Intervention

The second and last step repetitive task-oriented training involves correcting impaired motion patterns (3. Intervention). Throughout the session, intervention is provided by constant feedback on the patient's performance and assignment of new tasks if a patient does well. This step also involves reevaluating the stroke patient's stroke severity using the FMUE so new tasks can be set to continually meet Winstein's first two objectives for goal setting.

The problem with basing goal setting and intervention processes off of the FMUE, is the FMUE only evaluates stroke severity through static measurements, one joint at a time, and misses the motion strategy during dynamic functional tasks. This makes it difficult to set achievable goals for the patient if they are still having trouble completing the task assigned to them. Also, clinicians are not always available to provide feedback to a patient during the intervention process, which can lead to compensatory motion and frustration from slow or no progress.

Various human motion tracking technologies have been implemented to help correct upper extremity movements. In (Linden J., 2011), IMUs and haptic feedback from vibration motors were

placed on a subject to teach them the correct posture and bowing technique. Similar methods were used for stroke patients in (Huang M.C., 2011), where predefined therapist motions were masked in an on-screen jewel thief game controlled by Microsoft Kinect and pressure sensors.

Although these systems provide feedback to the user, the goals are predetermined. Also, if the subject deviates significantly from the target, personalized guidance cannot be given to them until they meet with a professional. There is yet a system that customizes the feedback for the patient and guides them through a hierarchical set of motion classes. A goal of this dissertation is to show that the sets of features derived from IMU data may be useful in prescribing therapies for stroke patients.

## 3.0 Aim 1: Define a Kinematic and Kinetic Based Feature Set From Which Assess and Predict Stroke Severity

In this chapter, a method will be presented that can be used to determine a given set of motion features that can classify stroke severity. Section 3.1 reviews previous studies that quantified differences in healthy and stroke motion patterns. Section 3.2 explains the rationale of defining the motion features using the information discussed in the previous section.

### 3.1 Theory and Methods

When an individual has a stroke, the three main motor deficits include hypertonia/reflexia (tightness of muscle tone/ overresponsive reflexes), dyscoordination (inability to effectively synchronize muscle effort from each limb to complete a task), and weakness (Levin M., 1996). These factors cause spatiotemporal irregularities in limb movement during daily living activities and can cause further injuries if not corrected. The current gold standard with which to evaluate the dyscoordination and weakness developed from stroke is the Fugl-Meyer Upper Extremity (FMUE) assessment. A patient's stroke severity is calculated by measuring their range of motion with a goniometer, grip strength by how they grasp and tug on an object (such as a soda can) and their speed of motion by measuring the time it takes for their index finger of their affected arm to touch their knee then bring their index to their nose. Woodbury (Woodbury M., 2013) performed a detailed analysis of the FMUE scoring system to define cutoff scores that set boundaries between upper extremity impairment (mild, moderate and severe) and described motor behavior at each

level. These cutoff scores were determined using the Rasch-Andrich step calibration (Andrich D., 1978; Linacre J., 1996). The results from this study are shown in Figure 1, with the cutoff FMUE scores and rank of FMUE task difficulty (the top task shown is defined as the easiest and bottom task is the hardest).

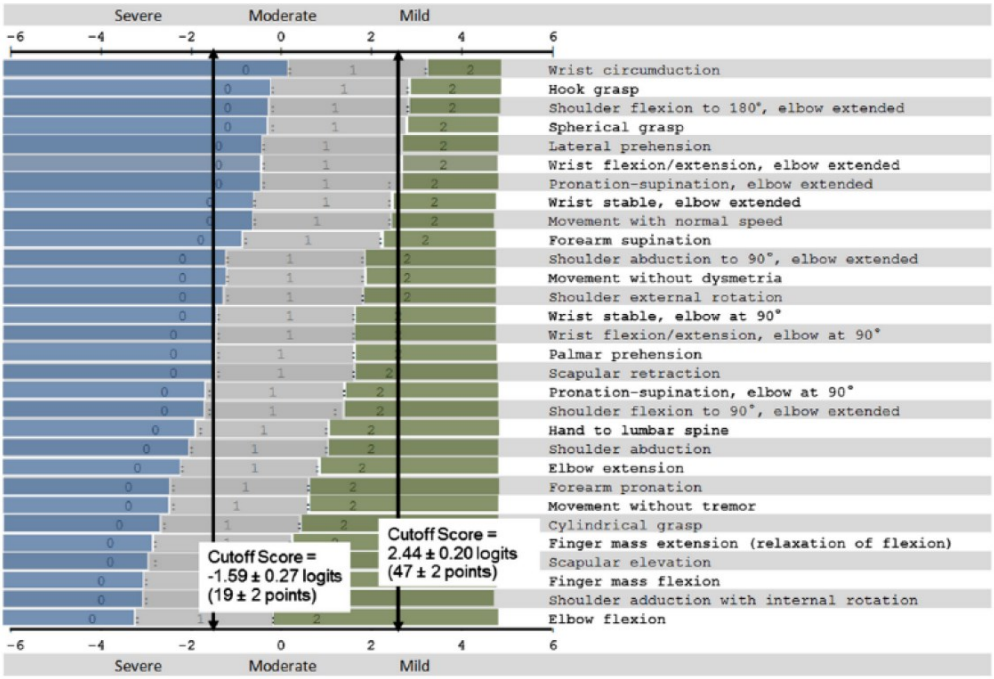



**Figure 1: FMUE with cutoff scores. The X-axis represents the Rasch logit measure. (Woodbury M. 2013)**

Although the FMUE scoring system measures motor impairment, it fails to observe dynamic compensatory motor strategies in between the start and end positions within the FMUE tasks.

Another limitation of the FMUE scoring system is its inability to distinguish the difference in the improvement of motor control from improvement in strength (Fugl-Meyer A.R., 1975; Gladstone D.J., 2002).The majority of the tasks defined in the FMUE scoring system primary requirement is on the strength of one or two joints to complete a task and can ignore the improvement of joint coordination that contributes to their score. With the full FMUE shown in Appendix A, only 3 task in section D of FMUE focus on coordination. By conducting kinematic and kinetic analysis during specially designed tasks that minimize strength requirements and restrict compensatory strategies, motor control recovery can be isolated and quantified (Beer R.F., 2007; Cortes J.C., 2017; Krebs. H.I., 2014).

With the rapid development of motion capture technologies, dynamic joint kinematic parameters can be measured at precise quantities. Murphy (Murphy M.A., 2011) used movement time, velocity, strategy (shape of velocity profile and location of peak), smoothness (number of tangential velocity peaks of the hand) and inter-joint coordination to identify differences in a reaching task between 19 healthy and 19 stroke participants (FMUE score $\geq$ 39). Their results showed that the smoothness, total movement time and peak angular velocity of the elbow were the best parameters to distinguish the difference between healthy and stroke participants. Although the kinematics were able to show movement differences in a binary case, their work wasn't compared to any standard clinical evaluations.

Yang (Yang Q., 2017) analyzed 6 kinematic parameters to define correlations with the FMUE scores. The features that showed significant relationships with the FMUE were the percentage of time to peak velocity (PTPV) and relative joint angle correlation (RJAC). The PTPV and RJAC monitored movement planning and joint coordination respectively. The results of this study are shown in Figure 2.

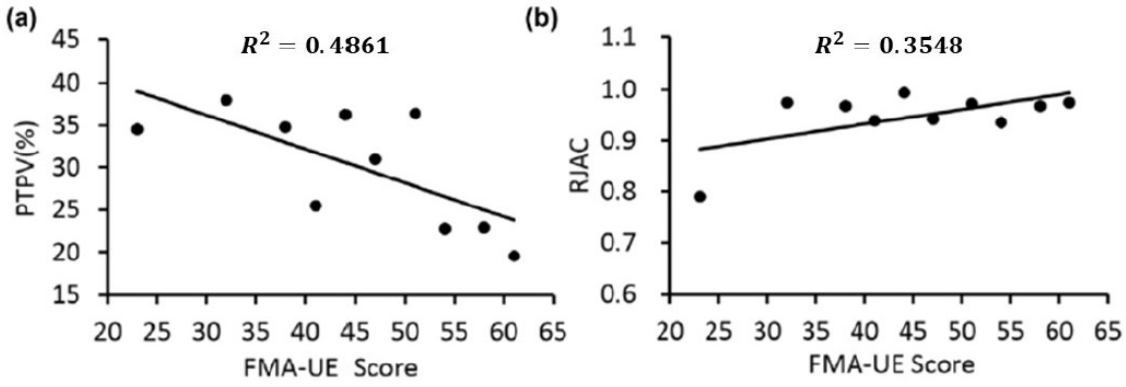



**Figure 2: Correlation between kinematic parameters and Fugl Meyer Assessment for Upper Extremity (FMA-UE) Score of 10 subjects with stroke at ID3.(a) Mean percentage time to peak velocity (PTPV). (b) Mean relative joint angles correlation (RJAC) (Yang Q. 2017)**

This study provides a foundation for refining parameters for clinical scores, but more variables need to be explored to find stronger correlations and to attain greater insight into the deterioration of joint coordination after stroke.

In this dissertation, similar kinematics can be obtained from the IMUs to correlate with the FMUE score and to get a detailed understanding of the dyscoordination of each stroke participant. The goal is to explore a large set of kinematic parameters (the complete list is provided in Appendix B) and use machine learning algorithms to find strong correlations between kinematics and stroke severity (as defined through FMUE scores).

The kinetic analysis helps to isolate strength improvement along with understanding how the central nervous system (CNS) performs trajectory planning. Within motion capture studies, inverse dynamic models have been incorporated to analyze the kinetics of an individual. Dewald (Dewald J., 2001) analyzed joint torques of control and stroke participants during reaching tasks across a table. The targets that were arranged in a circular pattern, forced the patient to generate shoulder flexion/extension (SF/SE), shoulder abduction/adduction (SAB/SADD), shoulder internal/external rotation (SIR), elbow flexion/extension (EF/EE) or a combination of these motions. Each patient performed an isometric (maximum torque was analyzed from load cell) or dynamic trial (torque analyzed through optical tracking along with an inverse dynamic model) throughout the study. The isometric results showed abnormal coupling (in terms of timing and magnitude) with the sets of [EF, AB, ER, SE] and [EE, AD, IR]. When shoulder abduction was incorporated in these tasks, there was a significant decrease in elbow extension, shoulder flexion and shoulder extension. During the dynamic trials, one severe stroke participant showed task-dependent elbow weakness. These results develop a foundation of motor control after stroke but to get a complete understanding, more joints need to be analyzed along with vertical motions that reflect more activities of daily living.

Raj (Raj S., 2018) went a step further by analyzing separate terms in his inverse dynamics model to examine changes in control of inter-segmental dynamics to understand dyscoordination caused by stroke. The torque terms calculated from the inverse dynamic model were grouped into passive torques (gravitational and interaction torque) and active torque (muscle torque) as described in (Hollerbach J., 1982). Control and stroke participants were asked to grab a soda can on a table while seated and lift the can to eye level. The results in this paper showed that stroke patients' motor strategy consisted of overcompensation of interaction torque. In other words,

stroke patients were highly dependent on forces from neighboring limbs to complete a reaching task, whereas a healthier individual's net torque demonstrated a higher dependency on muscle and gravity torque from a single joint to generate movement. It can also be seen in Figure 3 that the stroke severity and the change in torque have a positive correlation.

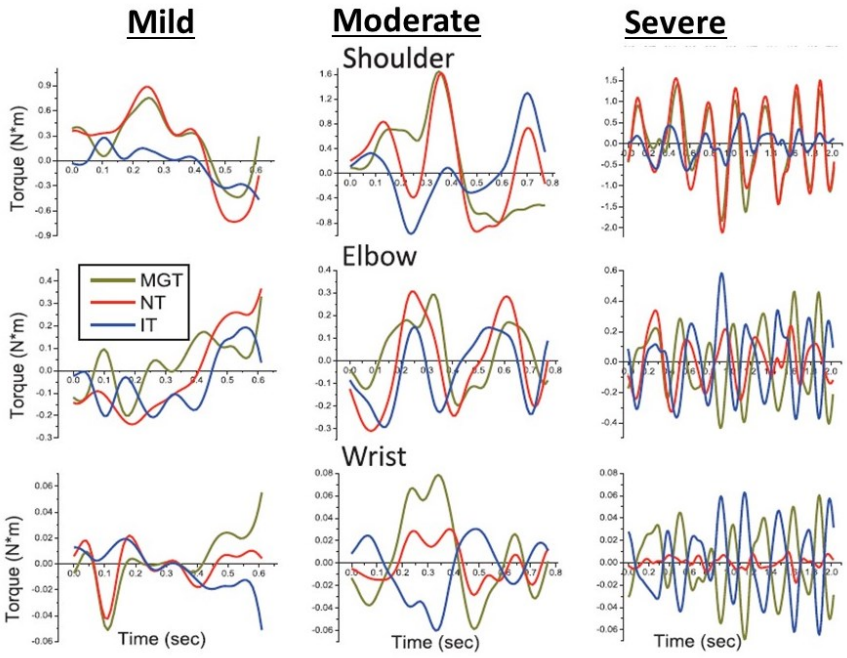



**Figure 3: Plots showing muscle and gravity torque (MGT), net torque (NT) and interaction torque (IT). Level 1-3 represents mild, moderate and severe stroke severities respectively.**

This distinct change in motor patterns has led researchers to search for the cause of this behavior. Flash (Flash T., 2012) explains that although the nature of how the CNS develops trajectory plans

is widely debated in the field, there is an agreement of a general motor hierarchy. The motor hierarchy consists of neural commands, muscle activations, joint motions, hand trajectory, and task goals. With muscle activation being the second level on the motor commands, joint torques will give researchers the closest understanding of the CNS.

Uno (Uno Y., 1989) was able to predict the hand trajectory in a two-dimensional plane by finding a trajectory that minimizes the change in the muscle torque at the shoulder and elbow joints. This follows the results in Raj's (Raj S., 2018) study in Figure 3 which illustrates a loss in trajectory planning from the CNS as the change in muscle torques increases. From looking at the muscle torque (input from CNS) and gravitational, interaction and net torques (output of CNS), a map of the CNS progression can be generated.

For this dissertation, one of the proposed kinetic parameters were chosen to be the mean square difference between the muscle torque and the focal point (referred to as net torque in (Raj S., 2018)), interaction, and gravitational torques of each joint. The result should show a strong correlation with the FMUE score and a map of CNS recovery by viewing the changing relationship between the input and output torques.

## 3.2 Motion Features Used to Analyze Stroke Severity

With the information from the literature mentioned in section 3.1, the total list of motion features that have been defined for this study are in Appendix B.

For the kinematic features, the one linear displacement that will be monitored is the average chest displacement of the participant. With stroke participants, it is expected for them to

compensate for their lack of range of motion of their arm by leaning forward during a reaching task.

The absolute phase difference between joint flexion angles will help to quantify the dyscoordination after stroke. For the other kinematic parameters such as angular position, velocity, acceleration, and jerk, there will be a continuation of Yang's work (Yang Q., 2017) to find which kinematics correlate with clinical scores such as the FMUE.

The focal point, muscle, interaction and gravity torques mentioned in (Raj S., 2018) have been calculated as the kinetic features for this study. Unlike in (Raj S., 2018), the inverse dynamics model has 10 degrees of freedom (DoF) that tracks forces along all 3 anatomical planes (sagittal, coronal and transverse planes) and will calculate these torques using IMUs. The max of each torque was chosen to see the compensation methods used from the interaction, gravity and focal point torques when a muscle torque is applied. The sum of the change of torque was chosen to monitor the damage in the CNS when a stroke occurs. By recruiting stroke participants with a range of stroke severities, a better understanding of how stroke affects the CNS commands to the muscles can be established.

From the large set of features, a support vector machine (SVM) along with feature selection algorithms will determine the most significant kinematic and kinetic features that can classify stroke severity. This process is explained in detail in chapter 5.0.

# 4.0 Motion Capture Data Acquisition

The purpose of this chapter is to discuss how the kinematic features were obtained from the IMU system that was used in this study. A brief discussion is also provided for the optical tracking system used to validate the IMU readings. This will help ensure that the IMU system behaved as expected.

Section 4.1 describes the experimental setup used for each participant to collect the motion features. The data acquisition methods for the IMUs and Optitrack systems are described in sections 4.2 and 4.3 respectively. Section 4.4 explains how the joint kinetics were calculated for the IMU system. The summary of findings for Aim 1 is discussed in section 4.5. Lastly, section 4.6, addresses how the measures of success were met in for Aim 1 in order to set up the machine learning stroke predictability in Aim 2.

## 4.1 Experimental Setup

For this study, kinematic and kinetic feature calculations require the capture of dynamic motions of the torso, shoulder, elbow and wrist joints. The patient performed a series of motions with an adjustable target system (shown in Figure 4) that promotes different joint coordination. Each trial began with the subject seated, the center target at eye level and the arm used for the task placed on an armrest. With the track system within the vertical and horizontal direction, the neutral position was readjusted for each patient.

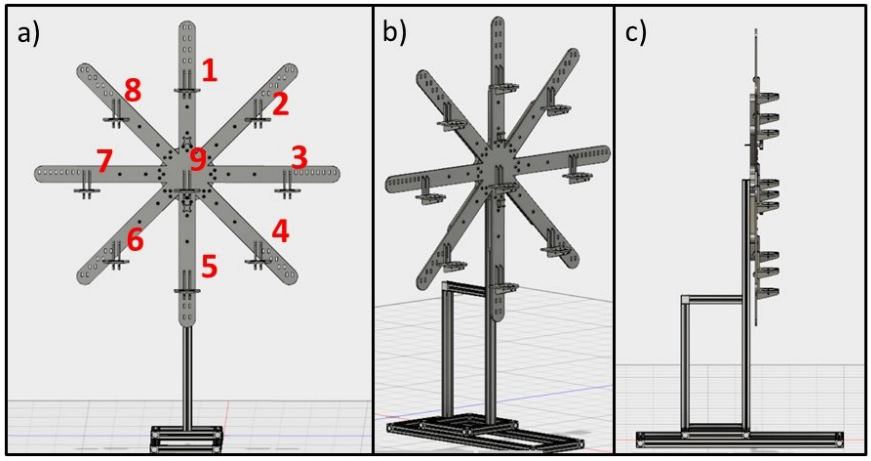



**Figure 4: CAD model of the target system used for stroke rehabilitation sessions (Shelf target attached for soda cans). The red numbers show the target assignments.**

During the motion analysis, the patient reached all 9 targets with their affected arm under 2 testing conditions. First, the target was placed at 40% and 80% of the patient's arm length for the short and far distance trials respectively. Second, three types of targets included a bullseye for pointing, a doorknob to twist and a soda can on a shelf to grab and lift. These three tasks will be referred to as the pointing, rotating, and grasping tasks for the remainder of this dissertation. Each spoke of the test rig, along with the center of the rig, holds only one target type, which is removable to quickly set up a different series of targets. With all the testing conditions (9 targets, 3 target types, 2 arm length positions), there were a total of 54 motions for each patient in a given test set.

A push-button was used to track the start and end of each trial and align the IMU and Optitrack system in time. The circuit that the button was attached to connected to a data acquisition (DAQ) board for the Optitrack system and a wireless dongle to transmit to the IMU system.

At the beginning of each trial, participants placed their hand on the push button and held it down until they were instructed to reach for a target. At the end of the trial, they had to return to the button and press down again. Figure 5 shows the button setup.

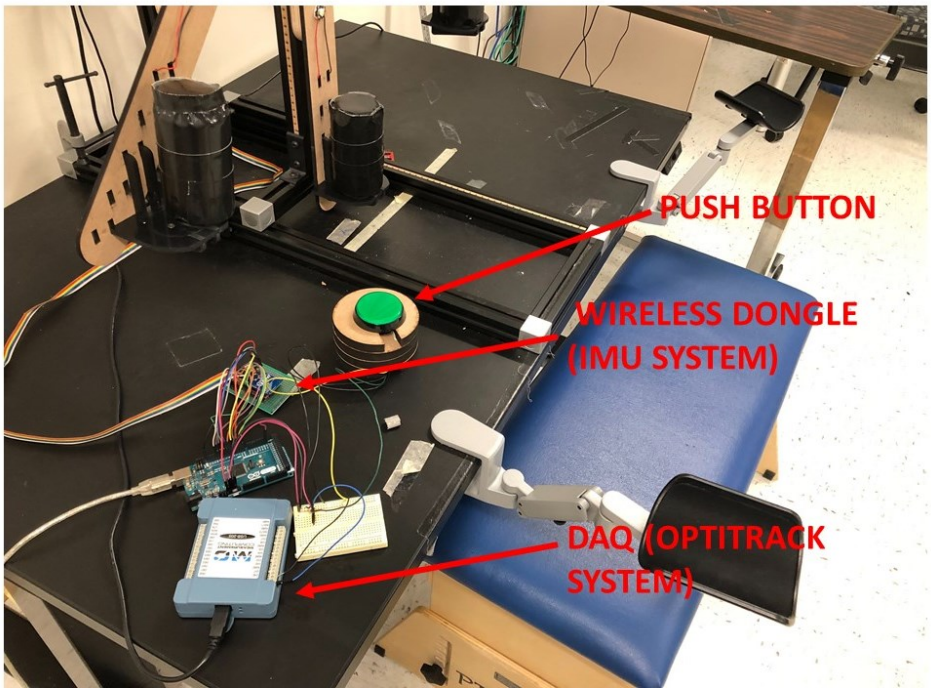



**Figure 5: Button setup for the experiment. The dongle sent the voltage signal to the IMU system while the DAQ sent the signal to the Optitrack system.**

Patients were recruited following the protocol in the Individualized Hand Improvement and Tracking System University of Pittsburgh Institutional Review Board (IRB) (Protocol number: PRO16070555). For this study, the goal was to recruit 10 healthy as control, and 10 mild stroke, and 10 moderate stroke patients. Stroke severity was identified by Dr. Amit Sethi using the FMUE scale.

## 4.2 Inertial Measurement Units: Theory and Methods

This section will explain how the joint kinematics are calculated from the IMU readings. The calculations follow the workflow shown in Figure 6.

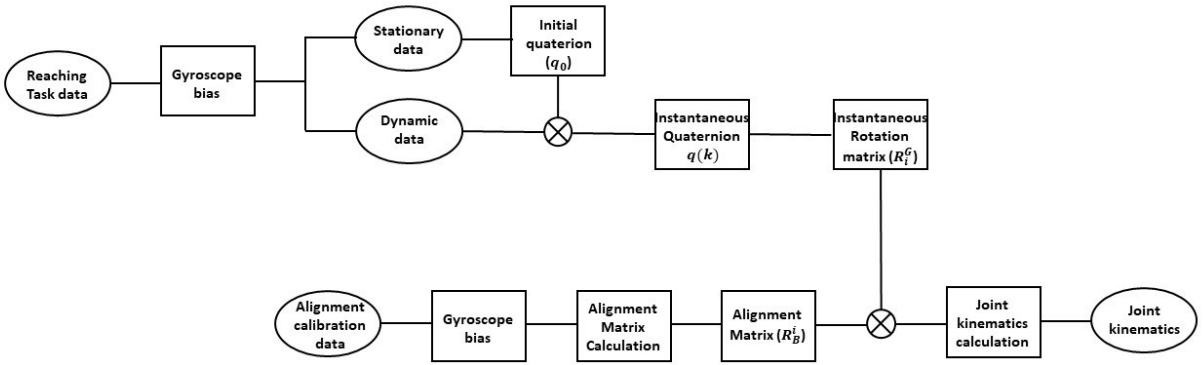



**Figure 6: Joint kinematics workflow.**

The flowchart in Figure 6 provides a summary of the methods discussed in this chapter to obtain the joint kinematics. Two motion trials need to be conducted to calculate the joint kinematics from the IMU. The alignment calibration data was first collected where patients moved their upper extremities in a pure joint rotation to define the transformation between body coordinate systems and the IMU coordinate system in the alignment matrix format (refer to section 4.2.1.3).

Once the alignment matrices were defined, the patient began the reaching task trial, where they used their affected hand to reach for a target shown in Figure 4. The patient is first asked to remain stationary to get the initial IMU orientation relative to the global coordinate system in the form of the initial quaternions. Then patients started the reaching task where the dynamic orientation was collected in the form of the instantaneous quaternions that were converted into an instantaneous rotation matrix (refer to section 4.2.1.1). By combining the dynamic readings with the alignment matrix, the IMU Euler angles were transformed into the anatomical frames to get the joint kinematics (refer to section 4.2.2).

The IMUs are placed into reflective marker plates by using a 3D printed case (The reflective markers will be described in detail in section 4.3.1). The full placement of IMUs on a participant is shown in Figure 7.

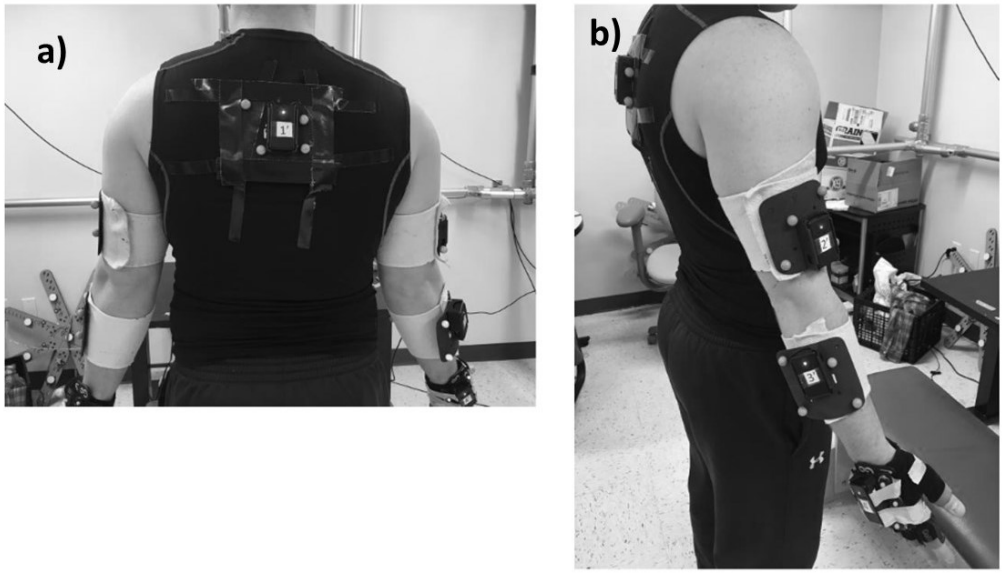



**Figure 7: a) Displays the back view of the participant where IMU1 is placed. B) Shows IMUs 2-4 placed on the upperarm, forearm and hand repectively**

A total of 10 joint angles were analyzed during the reaching tasks. The elbow was treated as a hinge joint so only the flexion angle was analyzed. For the torso, shoulder and wrist joints, they were treated as ball joints with 3 DoF each. Figure 8 shows all the body coordinate systems along with Table 1 that expresses the relationship between the Euler angles and the anatomical angles used in this dissertation.

**Table 1: List of Euler angles analyzed and their meaning**

| | **Euler angles** | **Anatomical angles** |
|---|---|---|
| **Torso** | $\phi_G^T, \theta_G^T \; and \; \psi_G^T$ | Lateral bending, axial rotation, and flexion/extension |
| **Shoulder** | $\phi_G^U, \theta_G^U \; and \; \psi_G^U$ | Abduction/ Adduction, rotation and flexion/extension |
| **Elbow** | $\psi_U^F$ | Flexion/extension |
| **Wrist** | $\phi_G^H, \theta_G^H \; and \; \psi_G^H$ | Flexion/extension, Forearm pronation/supination and deviation |

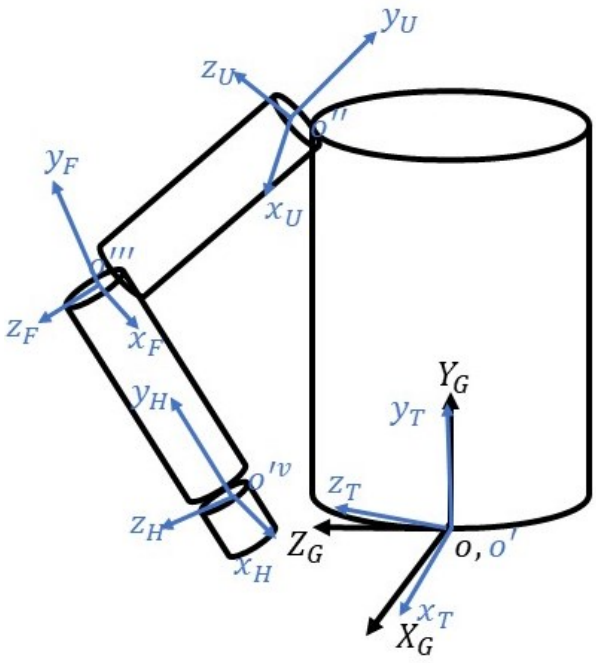



**Figure 8: The diagram shows the global (G), torso (T), upperarm (U), forearm (F) and hand (H) coordinate systems along a cylindical model of the body segments. Each "o" represents the origin of the respective coordinate system.**

The subscripts represent the initial coordinate system and the superscripts are the reference coordinate system that the measurement is being referenced to. For example, $\phi_1^2$ can be explained as the angle in the coordinate system 1 represented in coordinate system 2. *G*, *T*, *U*, *F*, and *H* represent the global, torso, upper arm, forearm and hand coordinate systems, respectively. To calculate joint angles in this dissertation, rotation matrices that relate an anatomical coordinate system to the global coordinate system were needed.

To further complicate the process, measurements were not made in either the anatomical or global coordinate systems. Instead, measurements were made in the IMU reference frames. That is, an IMU was attached to a body segment, but its coordinate system may not be aligned with the anatomical frame of that body segment. So, three coordinate systems were involved in calculations of motion for any given body segment: global coordinate system, anatomical (body) coordinate system, and IMU (sensor) coordinate system. To measure the angle between two body segments, five-coordinate systems were involved: the global coordinate system, the two anatomical coordinate systems, and the two IMU coordinate systems. An example of the coordinate systems used to calculate the elbow joint angle is shown in Figure 9.

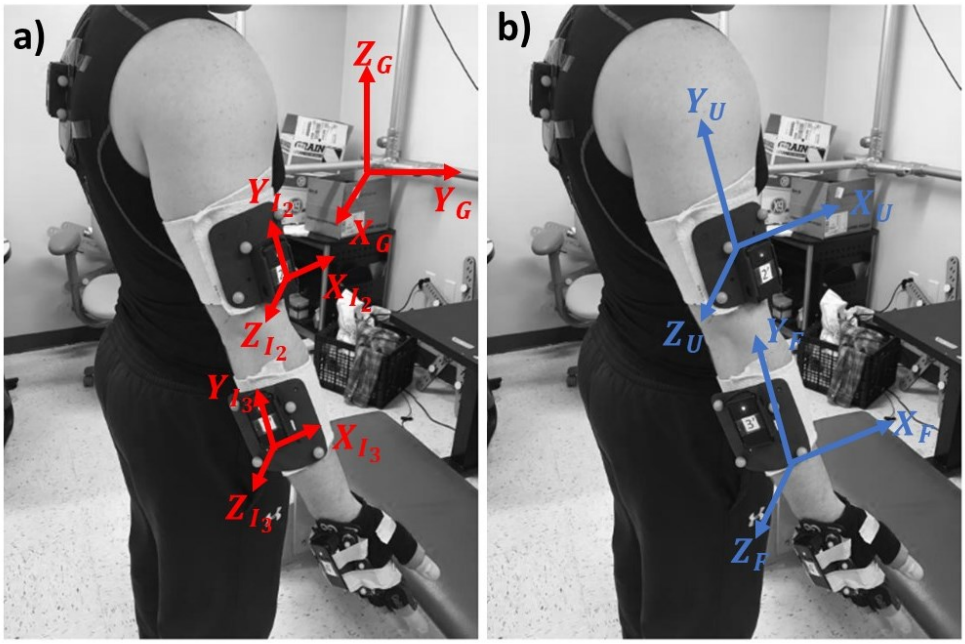



**Figure 9: a) shows the global coordinate system (G) and the IMU coordinate systems for IMU 2 (I2) and IMU 3 (I3) b) shows the two body coordinate systems for the upperarm (U) and forearm (F).**

In Figure 9, there were a total of five coordinate systems defined to calculate the elbow angle. In determining the elbow angle, the objective was to relate the orientation of the upper arm coordinate system to the forearm coordinate system in the fixed global frame. Since the two body coordinate systems can't be measured and updated directly, an indirect transformation approach was implemented. Instead of transforming from the body- to- global coordinate system directly, the inertial measurement unit (IMU) readings were transformed into the body coordinate system, then to the global coordinate system. This was done through one of the rotation matrix properties shown in equation (4-1).

$$R_B^G = R_i^G R_B^i \tag{4-1}$$

where $R_B^G$, $R_i^G$ and $R_B^i$ represent body- to- global, IMU- to- global and body- to- IMU rotation matrices respectively. The rotation matrices on the right side of equation (4-1) were calculated through IMU measurements given the techniques described next.

IMUs have been selected for this study due to their low cost, small size, and ease of operation. The IMUs used to track joint angles in this study consist of a three-axis accelerometer, gyroscope, and magnetometer. For the IMU to obtain its orientation relative to the global frame, a technique called sensor fusion is used to combine the readings of all the sensors. While the IMU is stationary, the accelerometer and magnetometer are used to obtain the initial orientation. The accelerometer tracks linear acceleration and captures the global gravitational force as a vertical global reference. While the magnetometer tracks the surrounding magnetic field, which is useful for determining a global horizontal reference direction (magnetic north). Once the initial orientation is calculated, the gyroscope signal that measures angular velocity is integrated to update the orientation.

In this dissertation, the rotation matrix method was used to implement the sensor fusion method and a way to relate to anatomical coordinate systems together to obtain a joint angle.

A rotation matrix is a cosine matrix that transforms a vector in one Euclidean space into another. This can be seen in equation (4-2).

$$v_b = R_i^b v_i \tag{4-2}$$

where $v_b$ represents a vector in the fixed body frame, $R_i^b$ is the rotation matrix from the inertial frame to the body frame and $v_i$ is the vector in the inertial frame. A visualization of the rotation matrix transforming the vector $v_i$ to $v_b$ is shown in Figure 10 and the calculation of a rotation matrix is shown in equation (4-3).

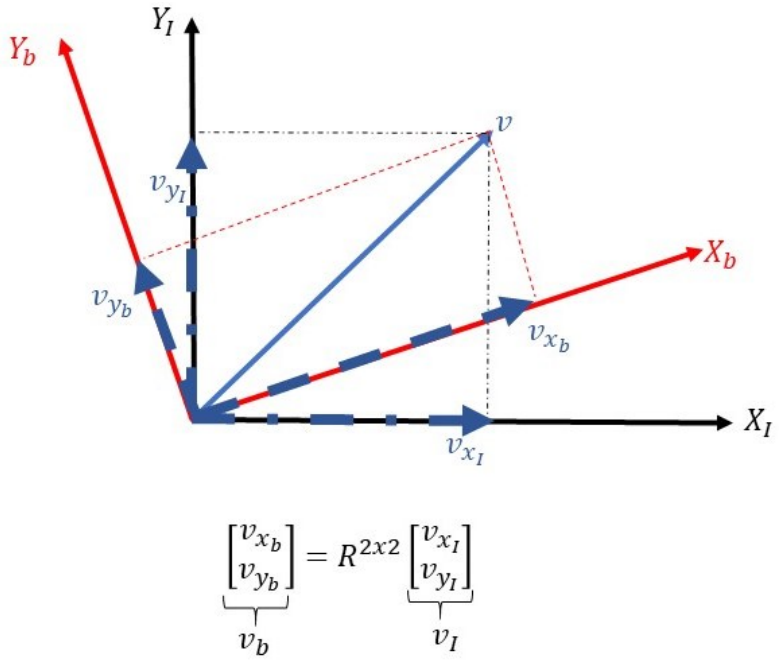



**Figure 10: Illustration showing the transformation a vector from the frame I to frame b.**

$$R_i^b = \begin{bmatrix} \cos(\theta_{x_b,x}) & \cos(\theta_{x_b,y}) & \cos(\theta_{x_b,z}) \\ \cos(\theta_{y_b,x}) & \cos(\theta_{y_b,y}) & \cos(\theta_{y_b,z}) \\ \cos(\theta_{z_b,x}) & \cos(\theta_{z_b,y}) & \cos(\theta_{z_b,z}) \end{bmatrix} \quad \textbf{(4-3)}$$

$\theta_{x_b,x}$ represents the angle between the $x_b$-axis of the inertial frame and $x$-axis of the fixed body frame (Diebel J., 2006). A problem that arises when creating the rotation matrix as described in equation (4-3) is commonly referred to as gimbal lock. Gimbal lock is a singularity problem where there is more than one solution to represent an orientation. To overcome this, quaternions were used to calculate a rotation matrix. A quaternion that represents a transformation from a fixed body frame to an inertial frame is shown in equations (4-4) – (4-5).

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos\frac{\theta}{2} & u_x sin\frac{\theta}{2} & u_y sin\frac{\theta}{2} & u_z sin\frac{\theta}{2} \end{bmatrix} \quad \textbf{(4-4)}$$

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad \textbf{(4-5)}$$

where $\theta$ is the rotation about the vector $\vec{u} = [u_x \quad u_y \quad u_z]$. To assure pure rotation without scaling, the quaternion was normalized to the magnitude of 1 (Diebel J., 2006; Kuipers J. B., 1999). To extend this to a 3-dimensional case or a transformation where rotations about 3 axes are needed, Euler angles can be used in quaternions. Euler angles represent the three rotations from the fixed body frame to an inertial frame. The most used ones are roll, pitch, and yaw. For this dissertation, the order of rotation will be a 123 sequence (sometimes referred to as an XYZ sequence). Where the rotation about the first axis is defined as the roll angle ($\phi$), the second defined by the pitch

angle ($\theta$) and the last defined as the yaw angle ($\psi$). A diagram of the order of rotation can be seen in Figure 11.

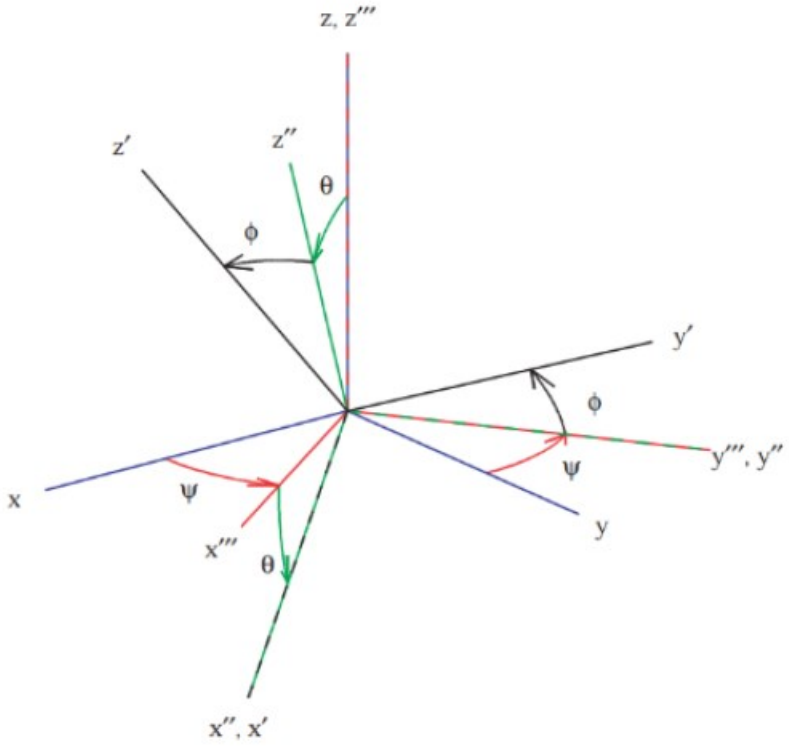



**Figure 11: Illustration of a 123 euler angle rotation sequence (Diebel J. 2006).**

This series of rotations is expressed mathematically in equations (4-6) – (4-9).

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \tag{4-6}$$

$$R_y(\theta) = \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \tag{4-7}$$

$$R_z(\psi) = \begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-8}$$

$$R_{xyz} = R_x(\phi)R_y(\theta)R_z(\psi) \tag{4-9}$$

where $c_{\emptyset/2} = \cos(\frac{\phi}{2})$ and $s_{\emptyset/2} = \sin(\frac{\phi}{2})$. Equation (4-9) represents the rotation of a vector about the X, Y and Z axes in a given coordinate system when it is initially aligned with a fixed reference coordinate system. The quaternion representation of the 3-dimensional rotation in equation (4-9) is shown in equation (4-10).

$$q = \begin{bmatrix} c_{\phi/2}c_{\theta/2}c_{\gamma/2} + s_{\phi/2}s_{\theta/2}s_{\gamma/2} \\ -c_{\phi/2}s_{\theta/2}s_{\gamma/2} + c_{\theta/2}c_{\gamma/2}s_{\phi/2} \\ c_{\phi/2}c_{\gamma/2}s_{\theta/2} + s_{\phi/2}c_{\theta/2}s_{\gamma/2} \\ c_{\phi/2}c_{\theta/2}s_{\gamma/2} - s_{\phi/2}c_{\gamma/2}s_{\theta/2} \end{bmatrix} \tag{4-10}$$

The quaternion in equation (4-10) can be used to construct a 123-sequence rotation as shown in equation (4-11).

$$R_i^b = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{4-11}$$

### 4.2.1 Calculating Rotation Matrices From the Inertial Measurement Unit Data

At each point in time for the collected data from the IMUs, a rotation matrix can be calculated that represents the rotation of the IMU in the global frame from the previous time point. The initial condition IMU to global rotation matrix was set using stationary data from the IMU. Then the initial rotation matrix was updated using the dynamic IMU data that created a series of instantaneous rotation matrices at each time step of the reaching task trial. The process used for calculating this sequence of rotation matrices is explained in this section.

#### 4.2.1.1 Calculation of the IMU-to-global rotation matrices Initial (Stationary) Rotation Matrix

The initial IMU-to-global rotation matrix was defined by measuring global forces in the IMU coordinate system while it is stationary. This ensured that gravity is the only acceleration being observed by the sensor and no acceleration is being contributed from the body segment the IMU is attached to. To get the initial orientation of the IMU coordinate system, the accelerometer was used to measure the direction of gravity. The initial Euler angles used for the quaternion calculation in equation (4-10) are calculated as shown in equations (4-12) – (4-16).

Roll angle:

for $\bar{A}_{IMU_z} < 0$:

$$\phi_0 = \tan^{-1}\left(\frac{\bar{A}_{IMU_y}}{\bar{A}_{IMU_z}}\right) \quad \textbf{(4-12)}$$

for $\bar{A}_{IMU_z} > 0$ and $\bar{A}_{IMU_y} > 0$:

$$\phi_0 = -\pi + \tan^{-1}\left(\frac{\bar{A}_{IMU_y}}{\bar{A}_{IMU_z}}\right) \quad \textbf{(4-13)}$$

for $\bar{A}_{IMU_z} > 0$ and $\bar{A}_{IMU_y} < 0$:

$$\phi_0 = -\pi + \tan\left(\frac{\bar{A}_{IMU_y}}{\bar{A}_{IMU_z}}\right) \quad \textbf{(4-14)}$$

Pitch angle:

$$\theta_0 = \tan^{-1}\left(\frac{\bar{A}_{IMU_x}}{\sqrt{\bar{A}^2_{IMU_y} + \bar{A}^2_{IMU_z}}}\right) \quad \textbf{(4-15)}$$

Yaw angle:

$$\gamma = 0 \quad \textbf{(4-16)}$$

where $\bar{A}_{IMU_x}$ represents the average IMU accelerometer reading along the x-axis while the participant is stationary. Figure 12 shows a basic illustration of how the $\phi_0$ is calculated using the accelerometer.

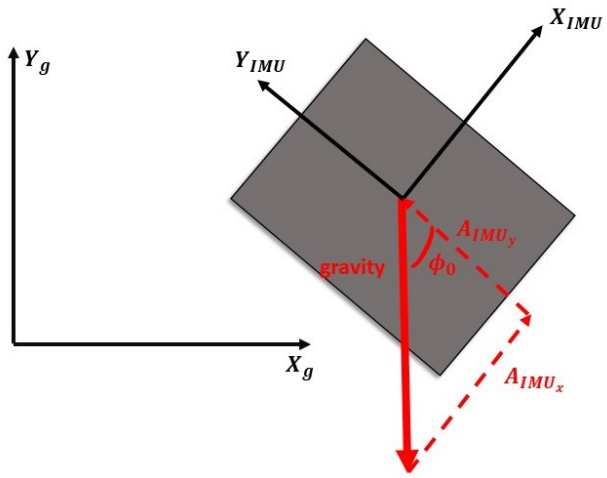



**Figure 12: Using the gravity projection onto the IMU axes, the initial orientation can be found.**

Since the inverse tangent function range is $[-\frac{\pi}{2}, \frac{\pi}{2}]$ (right side of a unit circle) the conditions for equations (4-12) – (4-14) are set to expand the range to $[-\pi, \pi]$ (all four quadrants of the unit circle). The yaw angle of the sensor coordinate system with respect to the global coordinate system is not measurable with the gravity method since gravity is a vertical reference vector within the global coordinate system (the axis of rotation for the yaw angle is set to be parallel with gravity). There is an infinite number of valid yaw angles for each roll and pitch angle solution. Practically speaking, for the analyses done in this dissertation, the global yaw angle is arbitrary (in other words, it doesn't matter if the patient is facing north or south, the motion data from the IMU is the same). The absolute yaw angle is not important, but the relative yaw angle (from one point in time to the next) is. Therefore, the yaw angle was initially and arbitrarily set to zero. Later in the

dissertation the yaw angle in the IMU data is related to the yaw angle defined in the Optitrack data for comparison purposes, but it is not required for operation of the wearable system. Changes in the yaw angle (as well as changes in pitch and roll) are determined for dynamic data from the angular gyro data.

Even though the absolute yaw angle is not necessary for IMU data analysis, it is important for comparison to Optitrack data. In many inertial measurement applications, the yaw angle would be calculated by the magnetometer. Due to the magnetic disturbance from the electronic adjustable table and other equipment in the lab, in this work, the absolute yaw angle was corrected in the Optitrack calibration phase which will be described in section 4.3.1.

#### 4.2.1.2 Dynamic Rotation Matrices

Once the initial rotation matrix or quaternion is obtained, future rotation matrices can be calculated from the dynamic IMU data. To update the IMU orientation in the global coordinate system, the IMU gyroscopes readings were integrated over time. The quaternions are updated as shown in equations (4-17) – (4-19).

$$q(k+1) = \exp\left(\frac{1}{2}\Omega(k)T_S\right) q(k) \tag{4-17}$$

$$q(k) = [q_0(k) \quad q_1(k) \quad q_2(k) \quad q_3(k)]^T \tag{4-18}$$

$$\Omega(k) = \begin{bmatrix} 0 & -\omega_x(k) & -\omega_y(k) & -\omega_z(k) \\ \omega_x(k) & 0 & \omega_z(k) & -\omega_y(k) \\ \omega_y(k) & -\omega_z(k) & 0 & \omega_x(k) \\ \omega_z(k) & \omega_y(k) & -\omega_x(k) & 0 \end{bmatrix} \tag{4-19}$$

where $k$ represents the current time step of the IMU reading, $T_s$ is the sampling period and $\omega_x(k)$ is the gyroscope reading about the IMU x-axis for that time step. Once the quaternion was updated, it was inserted back into equation (4-11) to update the IMU-to-global rotation matrix. Using this procedure, a set of IMU-to-global rotation matrices was obtained from the IMU-to-global frame for each instance of time for the motion studied.

#### 4.2.1.3 Alignment matrix: Body-to-inertial measurement rotation matrix

For the body-to-IMU rotation matrix, the participant was asked to perform motions about the given body frame. From this motion, IMU readings were related to the body frame to calculate columns of the rotation matrix. Each column in the body-to-IMU rotation matrix represents the body coordinate system axis relative to the IMU frame. This is expressed in equations (4-20) and (4-21).

$$R_B^i = \begin{bmatrix} \cos(\theta_{x_i,x_b}) & \cos(\theta_{x_i,y_b}) & \cos(\theta_{x_i,z_b}) \\ \cos(\theta_{y_i,x_b}) & \cos(\theta_{y_i,y_b}) & \cos(\theta_{y_i,z_b}) \\ \cos(\theta_{z_i,x_b}) & \cos(\theta_{z_i,y_b}) & \cos(\theta_{z_i,z_b}) \end{bmatrix} = [X_B^i \quad Y_B^i \quad Z_B^i] \tag{4-20}$$

$$X_B^i, Y_B^i, Z_B^i \in \mathbb{R}^{3x1} \tag{4-21}$$

where $X_B^i$, $Y_B^i$, and $Z_B^i$ are the $x$, $y$ and $z$ body axes represented in the IMU frame. When participants were asked to perform a motion to calculate the alignment matrix, it is assumed that the IMU readings were capturing motions being performed purely in a plane perpendicular to a given body axis. In other words, when the IMU moves purely in a plane perpendicular to a given body axis it becomes projected onto the IMU frame. Figure 13 shows an example of a participant performing the alignment calibration for the body Z-axis.

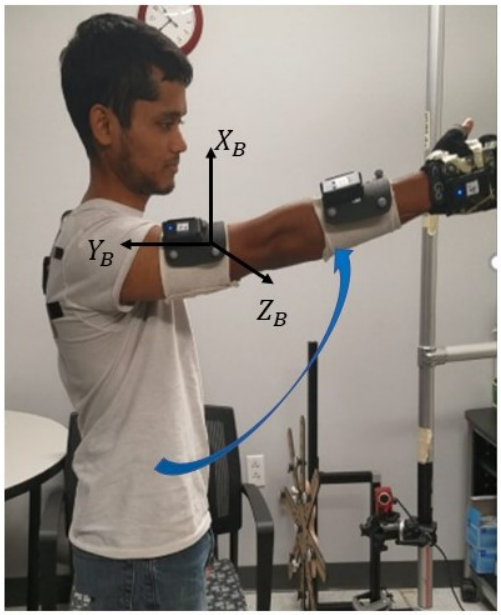



**Figure 13: Shoulder flexion was used to defined the body Z axis for the 3 IMUs attached to the arm.**

The alignment matrix calculation process described in the rest of the section was applied similarly for all four IMUs attached to the participant, but of course with appropriate body segments being moved for each. The descriptions for the movement along anatomical planes and axes that defined the alignment matrix is shown in Figure 14.

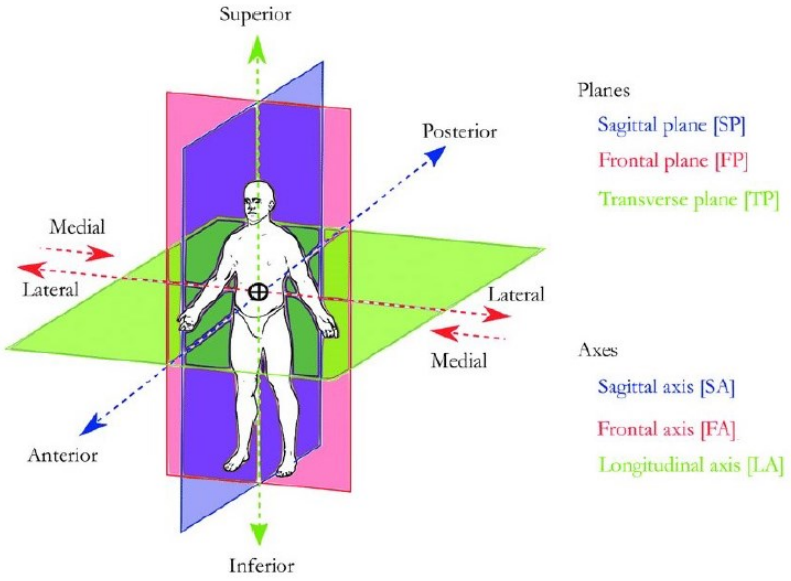



**Figure 14: Image showing the definitions of anatomical planes and axes (Richard V., 2016).**

To define the second column of the alignment matrix for each IMU, participants were asked to stand still for 5 seconds to collect the stationary accelerometer signal. The longitudinal axis (body Y-axis) was assumed to be parallel with gravity. The calculation for the second column is shown in equations (4-22) and (4-23).

$$\hat{Y}_B^i = \hat{R}_{*2} = \frac{\text{sum}(A_{IMU})}{\|\text{sum}(A_{IMU})\|} \tag{4-22}$$

$$\text{sum}(A_{IMU}) = \left[\sum_{k=1}^{N} A_{IMU_x}(k) \quad \sum_{k=1}^{N} A_{IMU_y}(k) \quad \sum_{k=1}^{N} A_{IMU_z}(k)\right]^T \tag{4-23}$$

where $\hat{Y}_B^i$ and $\hat{R}_{*2}$ represent the estimate of the body *y*-axis and the second column of the IMU to body rotation matrix, respectively. For the summation term, *N* is the last time step in the stationary period. Next, the participants were asked to rotate each body segment along a plane perpendicular to the frontal horizontal axis (body Z-axis).

For the IMU attached to the back, participants were asked to perform 5 repetitions of torso flexion. For the IMUs attached to the upper arm, forearm, and hand, the participant was asked to perform 5 repetitions of shoulder flexion at 90 degrees. The gyroscope signal was used to record both motions. Participants performed 5 rotations to account for cases where they did not purely flex across the given joint and would cause the definition of the body coordinate system to have non-orthogonal axes. Each trial creates an estimate for the body Z-axis making a total of 5 separate estimates.

The third column of the rotation matrix was defined by the positive rotations of an upper extremity limb (from resting position to maximum flexion). This makes sure that the Z-axis is defined along the transverse axis from the body segment. The calculation for the third column is shown in equations (4-24) – (4-25) below.

$$\hat{Z}_B^i(m) = \hat{R}_{*3}(m) = \frac{\text{sum}(\omega_{IMU})}{\|\text{sum}(\omega_{IMU})\|} \tag{4-24}$$

$$where, \qquad \text{sum}(\omega_{IMU}) = \left[\sum_{k=1}^{N} \omega_{IMU_x}(k) \quad \sum_{k=1}^{N} \omega_{IMU_y}(k) \quad \sum_{k=1}^{N} \omega_{IMU_z}(k)\right]^T \tag{4-25}$$

where $\hat{Z}_B^i(m)$ and $\hat{R}_{*3}(m)$ represents the "*m*" number of estimates of the body z-axis and the third column of the IMU to body rotation matrix, respectively. The number of estimates "*m*" was

determined by the number of flexion/extension repetitions that were performed during this calibration procedure. For the summation term, N is the time step where the participant has reached maximum flexion. Maximum flexion was defined when the magnitude of the IMU angular velocity returned to zero after the motion was first initiated.

The estimates for the first column of the rotation matrix were then defined by the cross product of the second and third columns (Y and Z axes). This calculation is shown in equations (4-26) – (4-27).

$$\hat{X}_B^i(m) = \hat{R}_{*1}(m) = \frac{\hat{R}_{*1}(m)}{\left\|\hat{R}_{*1}(m)\right\|} \tag{4-26}$$

$$where, \qquad \hat{R}_{*1}(m) = \hat{R}_{*2} \times \hat{R}_{*3}(m) \tag{4-27}$$

Given the procedure described in equations (4-22) – (4-27), there are initially 5 IMU- to- body rotation matrix estimates for each segment. Since participants may not have rotated purely along the frontal horizontal axis, the columns may not be orthogonal to each other. To ensure orthogonality a correction procedure was developed.

Since it is likely that the $\hat{X}_B^i(m)$ and $\hat{Y}_B^i$ axes were not orthogonal to the $\hat{Z}_B^i(m)$ axes due to the inconsistent flexion trials, the matrix theory properties from equations (4-28) – (4-29) were used to correct these axes.

$$\|r_i\| = \left\|c_j\right\| = 1 \; for \; (i = 1,2,3 \; and \; j = 1,2,3) \tag{4-28}$$

$$r_i \cdot r_j = 0 \; and \; c_i \cdot c_j = 0 \; for \; (i = 1,2,3 \;\; j = 1,2,3 \; and \; i \neq j) \tag{4-29}$$

To make each estimate of the $\hat{X}_B^i(m)$ axis orthogonal to the $\hat{Z}_B^i(m)$ axis, the elements in the first column were modified with multiplier terms as shown in equation (4-30) and then applied to equations (4-31) – (4-32) were applied to ensure orthogonality between the body axes (Zhang, 2014).

$$\tilde{R}_{*1}(m) = [\hat{R}_{11}(m)a \quad b \quad \hat{R}_{31}(m)a] \tag{4-30}$$

$$\left\|\tilde{R}_{*1}(m)\right\| = \left(\hat{R}_{11}(m)a\right)^2 + b^2 + \left(\hat{R}_{31}(m)a\right)^2 = 1 \tag{4-31}$$

$$\tilde{R}_{*1}(m) \cdot \hat{R}_{*2} = \hat{R}_{11}(m)a\hat{R}_{12} + b\hat{R}_{22} + \hat{R}_{31}(m)a\hat{R}_{32} = 0 \tag{4-32}$$

where *a* and *b* are substitute terms to make sure equations (4-28) and (4-29) are true such that the column is unity and orthogonal to the third column. With the system of equations in (4-31)and (4-32), the solution for the a and b unknowns are given below.

$$a = \sqrt{\frac{R_{22}^2}{R_{11}^2R_{22}^2 + (R_{11}R_{12} + R_{31}R_{32})^2 + R_{31}^2R_{22}^2}} \tag{4-33}$$

$$b = \frac{-a(R_{11}R_{12} + R_{31}R_{32})}{R_{22}} \tag{4-34}$$

Substituting these values back into equation (4-32) creates a new set of estimates ($\tilde{R}_{*1}(m)$) for the first column of the body-to-IMU rotation matrix that is now orthogonal to the second column. This procedure was repeated for the third column with respect to the second column of the IMU to body rotation matrix to ensure that these are also orthogonal to the second column.

The last step was to make sure the rotation matrix defined the transformation to the body axis where the body X-axis was parallel with the horizontal sagittal axis and the body Z-axis was parallel with the frontal horizontal axis. To accomplish this, the offset rotation to align the body Z axes in the right direction must be calculated. This was done by using the first body Z-axis as a reference vector and then finding the average angle between the reference vector and the other body Z axes estimates along with the reference vector and the estimated body X axes. The calculation is shown in equations (4-35) – (4-38).

$$Z_{ref} = \widetilde{R}_{*3}(m) \tag{4-35}$$

$$\theta_{zz}(m) = \frac{Z_{ref} \cdot \widetilde{R}_{*3}(m)}{\left\|Z_{ref}\right\|\left\|\widetilde{R}_{*3}(m)\right\|} \tag{4-36}$$

$$\theta_{zx}(m) = \frac{Z_{ref} \cdot \widetilde{R}_{*1}(m)}{\left\|Z_{ref}\right\|\left\|\widetilde{R}_{*1}(m)\right\|} \tag{4-37}$$

$$\theta_{est} = \frac{\sum_{m=1}^{N} \theta_{zz}(m) + \theta_{zx}}{2 * N} - \frac{\pi}{4} \tag{4-38}$$

where $Z_{ref}$ is the reference body Z-body axis from the first estimate m, $\theta_{zz}(m)$ is the angle between the body Z-body axis reference and the estimate m of the third rotation matrix column, and $\theta_{zx}(m)$ is the angle between the body Z-axis body reference and the estimate m of the first rotation matrix column. The $Z_{ref}$ vector is then rotated about the body Y-body axis using the equations (4-39) – (4-41) below.

$$R_{\hat{Y}_b}(\theta_{est})$$

$$= \begin{bmatrix} \hat{R}_{12}^2(1-C_{\theta_{est}}) + C_{\theta_{est}} & \hat{R}_{12}\hat{R}_{22}(1-C_{\theta_{est}}) - \hat{R}_{23}S_{\theta_{est}} & \hat{R}_{12}\hat{R}_{23}(1-C_{\theta_{est}}) + \hat{R}_{22}S_{\theta_{est}} \\ \hat{R}_{12}\hat{R}_{22}(1-C_{\theta_{est}}) + \hat{R}_{23}S_{\theta_{est}} & \hat{R}_{22}^2(1-C_{\theta_{est}}) + C_{\theta_{est}} & \hat{R}_{22}\hat{R}_{23}(1-C_{\theta_{est}}) - \hat{R}_{12}S_{\theta_{est}} \\ \hat{R}_{12}\hat{R}_{23}(1-C_{\theta_{est}}) - \hat{R}_{22}S_{\theta_{est}} & \hat{R}_{22}\hat{R}_{23}(1-C_{\theta_{est}}) + \hat{R}_{12}S_{\theta_{est}} & \hat{R}_{23}^2(1-C_{\theta_{est}}) + C_{\theta_{est}} \end{bmatrix} \quad \textbf{(4-39)}$$

$$Z_{cor} = R_{Y_b}(\theta_{est}) \cdot Z_{ref} \quad \textbf{(4-40)}$$

$$Z_{cor} = \frac{Z_{cor}}{\|Z_{cor}\|} \quad \textbf{(4-41)}$$

where $R_{\hat{Y}_b}(\theta_{est})$ is the rotation matrix for a rotation of $\theta_{est}$ about the $\hat{Y}_b$ axis and $Z_{cor}$ is the corrected body Z-axis that is now parallel with the frontal horizontal axis. The remaining columns were calculated and assembled in the IMU-to-body rotation matrix as shown in equations (4-42) – (4-44).

$$X_{cor} = \hat{R}_2 \times Z_{cor} \quad \textbf{(4-42)}$$

$$X_{cor} = \frac{X_{cor}}{\|X_{cor}\|} \quad \textbf{(4-43)}$$

$$R_i^b = [X_{cor} \quad \hat{R}_2 \quad Z_{cor}] \quad \textbf{(4-44)}$$

where $X_{cor}$ represents the corrected body X-axis that is now parallel with the horizontal sagittal axis. It is important to note that in this dissertation the relationship between the body and IMU was assumed to be constant after the calibration is done. The five alignment matrix estimates along with the corrected alignment matrix are shown in Figure 15.

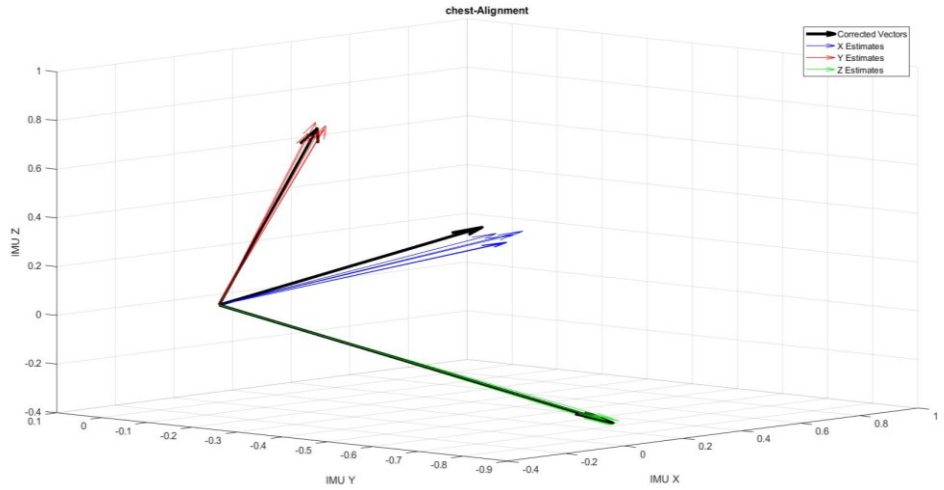



**Figure 15: Shows the 5 estimates of each body axes relative to the IMU coordinate system. The thick black line represents the final corrected body axes.**

### 4.2.2 Joint Kinematics Calculation

After finalizing the calculations of the IMU-to-global and IMU-to-body rotation matrices, the rotation matrix that relates two body coordinate systems were related to each other to get the joint angle. Referring to Figure 9 and expanding on equation (4-1) the transformation from one body coordinate system to another is calculated in equation (4-45).

$$R_U^F = R_{I_3}^F R_G^{I_3} R_{I_2}^G R_U^{I_2} \quad \textbf{(4-45)}$$

$R_U^F$ represents the rotation matrix from the upper arm to the forearm coordinate system. $I_2$ and $I_3$ are the coordinate system of the IMUs attached to the upper arm and forearm respectively. Once the rotation matrix between these two body coordinate systems has been calculated, the Euler angles were extracted from $R_U^F$ rotation matrix as follows.

$$\phi_U^F = atan2\left(\frac{R_{23}}{R_{33}}\right) \quad \textbf{(4-46)}$$

$$\theta_U^F = -\text{asin}\,(R_{13}) \quad \textbf{(4-47)}$$

$$\psi_U^F = atan2(R_{12}, R_{11}) \quad \textbf{(4-48)}$$

Since these angles represent the movement of the elbow joint, which is assumed to be a hinge joint, only the elbow flexion angle is needed. For these Euler angles, the rotation about the Z-axis or yaw angle between the two frames of reference represents the flexion angle.

Once the Euler angles were obtained, the joint velocities and accelerations are needed for the kinetic calculations which will be described in section 4.4. The joint velocity is calculated in equation (4-49).

$$\omega_U^F = R_G^F R_U^G \omega_U \quad \textbf{(4-49)}$$

$$where, \omega_U = R_{I_2}^U \omega_{I_2} \quad \textbf{(4-50)}$$

$$\omega_U^F = \begin{bmatrix} \dot{\phi}_U^F \\ \dot{\theta}_U^F \\ \dot{\psi}_U^F \end{bmatrix} \quad \textbf{(4-51)}$$

$\omega_U$ represents the angular velocity vector for the upper arm body coordinate system and $\omega_U^F$ is the angular velocity of the upper arm relative to the forearm. Each element within the $\omega_U^F$ represents the time derivative of the Euler angles defined in equations (4-46) – (4-48). A comparison of using the method in equation (4-49) versus taking the derivative of the Euler angles can be seen in Figure 16.

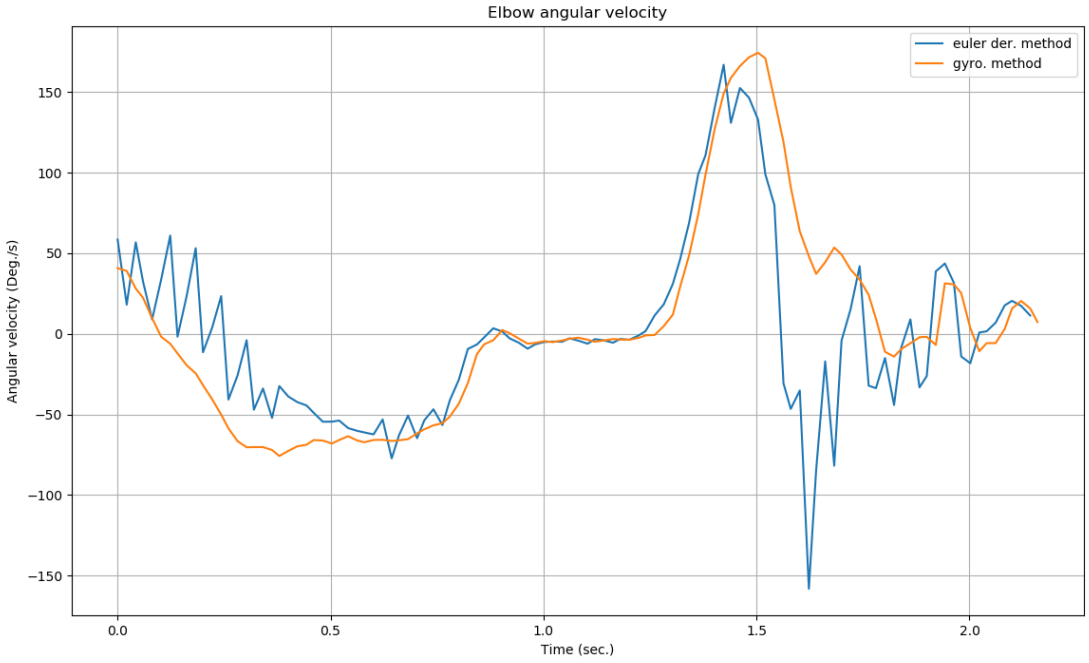



**Figure 16: Euler angle derivative vs the gyroscope method described in equation (4-49).**

It can be seen in Figure 16, that taking the direct time derivative of the Euler angle adds more noise to the signal vs transforming the angular velocity from the gyroscope signal into the body frames using equation (4-49).

The method described in this section was applied to the other angular joint velocities. For the angular acceleration, the time derivative of the angular velocity was calculated. In future work, the distance of the IMU from the joint can be used to get an angular acceleration directly from the accelerometer signal instead.

## 4.3 Optical Tracking System: Theory and Methods

Optical motion tracking systems are the current gold standard for telerehabilitation studies. The system tracks a participant's motion using reflective markers (passive system) or blinking LEDs at predetermined frequencies (active system) to track limbs in a 3-dimensional space. With passive marker systems, the camera can track a reflective marker from the light sources in the room and/or light sources from the camera itself. Since all the markers are exactly alike, clusters of markers fixed to a rigid base are normally used on body segments, oriented in unique patterns, so that the optical tracking system can monitor a cluster of markers instead of a single marker.

Active marker systems avoid the problems faced from passive markers by having LED markers flash at unique frequencies. This eliminates confusion over distinguishing markers from each other and it eliminates the need for adjusting the lighting in the room if certain markers aren't as reflective. One fault in this system is the need to have a battery pack(s) to power the LEDs along with troublesome wiring around the rigid bodies.

For both systems, the biggest drawback is the limited space that motion can be tracked, the steep learning curve to operate, and the expense of the system.

For this dissertation, the Optitrack (NaturalPoint, Inc) optical tracking system (passive system) was used to validate the IMU joint estimations. This was to validate the accuracy of the kinematic and kinetic features extracted from IMU data for when the IMUs are used outside of the laboratory setting.

### 4.3.1 Optical Tracking Kinematics

The cameras for this study were orientated as shown in Figure 17.

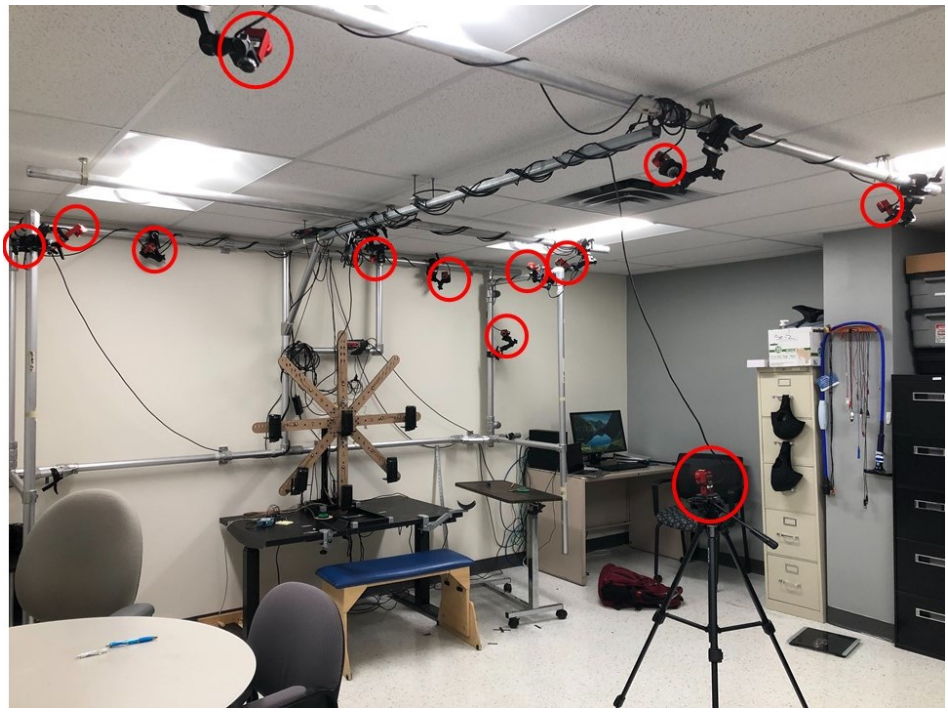


**Figure 17: Shows the camera placement for the experiment. Cameras are circled to highlight their locations**.

The marker plates shown in Figure 7 in section 4.2.1 were tracked in Optitrack's Motive software where each marker's position and orientation in the 3-dimensional space are recorded. This information was then sent to a third-party software called Motion Monitor (Innovative Sports Training, Inc.). Motion Monitor was used to generate a model of the participant's upper extremity, calculate the joint kinematics using the Optitrack data and export it into a text file for further analysis.

To create a model of the participant's upper extremities, a stylus with reflective markers attached was used to locate anatomical points relative to the marker plates. The length of each limb was calculated by placing the stylus on the distal and proximal joints surrounding the limb. Then the body coordinate systems were defined using the same distal and proximal points along with an anterior point. Table 2 shows where the stylus was placed for each limb and Figure 18 shows the visual output to this model.

**Table 2: Anatomical landmarks to define the upper extremity model**

| BODY SEGMENT | PROXIMAL LANDMARK | DISTAL LANDMARK | ANTERIOR LANDMARK |
|---|---|---|---|
| **Back (Torso)** | T12-L1 Joint | C7-T1 Joint | Right Acromion |
| **Left Upper Arm** | Left Acromion | Antecubital Fossa | Right Acromion |
| **Left Forearm** | Medial Epicondyle | Ulnar Styloid | Lateral Epicondyle |
| **Left Hand** | Underside of Wrist, at Radiocarpal Joint | Underside of 1st Knuckle of 3rd Phalanx | Tip of 1st Disyal Phalanx (Thumb) |
| **Right Upper Arm** | Right Acromion | Antecubital Fossa | Left Acromion |
| **Right Forearm** | Medial Epicondyle | Ulnar Styloid | Lateral Epicondyle |
| **Right Hand** | Underside of Wrist, at Radiocarpal Joint | Underside of 1st Knuckle of 3rd Phalanx | Tip of 1st Distal Phalanx (Thumb) |

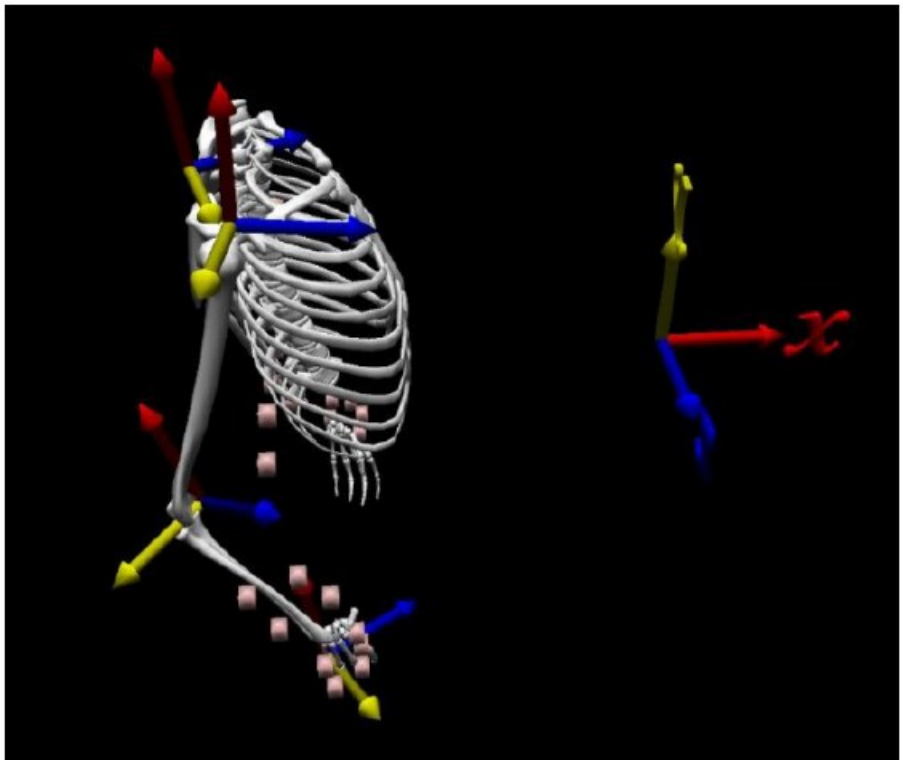


**Figure 18: Screenshot of the upper extremity model in Motionmonitor. The segment lengths of the skeleton are proportional to the participant. The body coordinate systems defined in Table 3 are shown along with the global coordinate system that is not attached to the model.**

With the complete upper extremity model, Motion Monitor was used to calculate the joint kinematics using the Euler angles between body coordinate systems as described in section 4.2.2.

It is important to note that the yaw angle extracted in the Motion Monitor software was used to correct the yaw angle for the IMUs. As described earlier, this was done because the IMU magnetometers could not be used to determine the absolute yaw angle due to the high amount of

magnetic interference from the metal frame used to support the Optitrack cameras, the motorized desk, and other objects in the laboratory space. While normally the absolute yaw angle from IMU data is not important, it is needed here for comparison to between the Optitrack- and IMU-derived motion results. In the future work, a Kalman filter similar to the one mentioned in (Sabatini A.M. , 2006) can be applied to set a level of dependency of the magnetometer measurements in the presence of magnetic disturbance.

## 4.4 Inverse Dynamics Model for Joint Kinetics

To calculate the kinetics of each participant, an inverse dynamics model was derived. An inverse dynamics model uses kinematics and mass properties of segments to calculate joint torques and forces. For this dissertation, the Euler-Lagrange approach was used to derive the governing equations. The Lagrange equation is defined in equations (4-52) – (4-54).

$$L = KE - PE \tag{4-52}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{4-53}$$

$$q = [\phi_T \quad \theta_T \quad \psi_T \quad \cdots \quad \phi_W \quad \theta_W \quad \psi_W] \tag{4-54}$$

where $L$ is the Lagrange variable that is equal to the kinetic energy (KE) of the system minus the potential energy (PE). The Euler angles were used as the generalized coordinates in (4-54). To define the kinetic and potential energy terms, a few assumptions had to be made. The first being

that each segment was assumed to be a cylinder. The height, mass, and center of mass distance from the proximal joint (assumed placement of IMUs) were derived from an anthropometric model for each segment (Winter D.A., 2009). The radius was calculated from the parameters recorded in (Gupta M., 2017) where the ratio of the circumference of each segment over their height was obtained from the 50th percentile of US men. Before each testing session, the height and weight of the participant were taken to get an estimate of the length and weight of the limbs. These assumptions are listed in Table 3 along with the anthropometric model from (Winter D.A., 2009) in Figure 19.

**Table 3: Segment measurement assumptions, where H and M are the total height (in meters) and mass (in kg) of the participant.**

| | Segment length (m) | Segment mass (m) | Segment mass (m) | Center of mass location from proximal (m) |
|---|---|---|---|---|
| **Torso** | 0.288H | 0.129H | 0.521M | 0.5·segment length |
| **Upperarm** | 0.186H | 0.0311H | 0.029M | 0.436·segment length |
| **Forearm** | 0.146H | 0.0274H | 0.017M | 0.43·segment length |
| **Hand** | 0.108H | 0.0169H | 0.006M | 0.506·segment length |

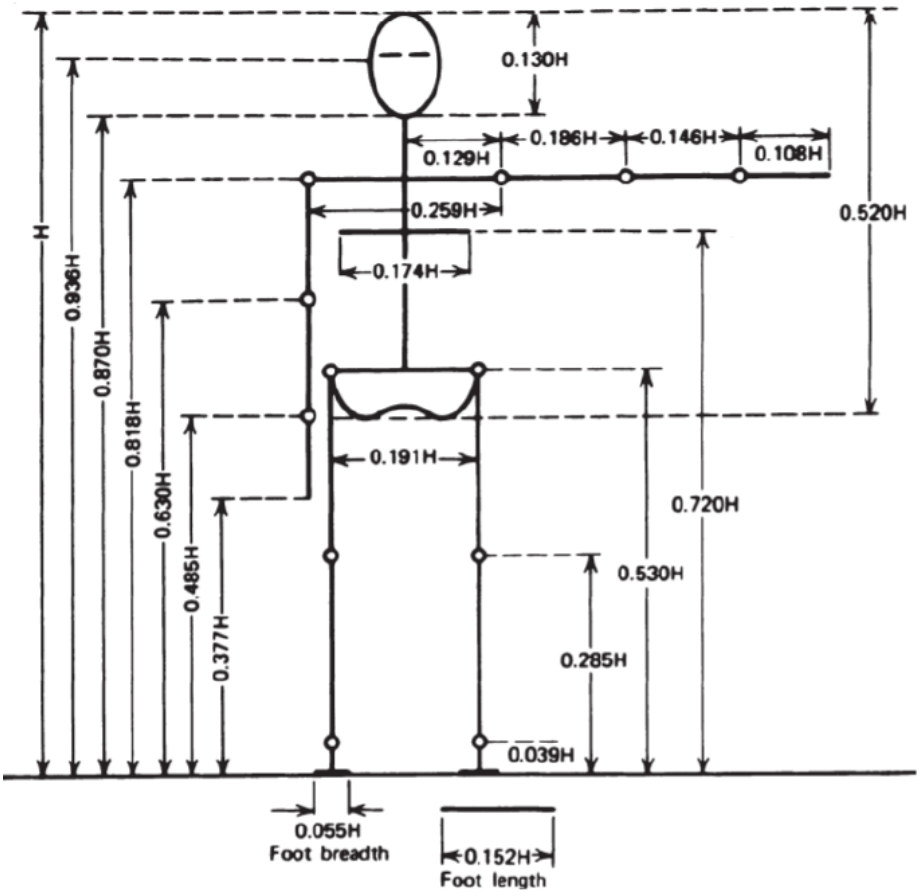



**Figure 19: Anthropometric model from (Winter D.A. 2009). Each segment length is a ratio of the person's height (H).**

The second assumption that was made was that the body coordinate system was parallel with the principal axes of inertia to simplify the inertia matrix (***I***). With these assumptions set, the position and velocity of the center of mass of each limb were represented in the global frame and used in the kinetic and potential energy equations.

$$KE = \frac{1}{2} m_T \, {}^{G}\vec{v}_{olc_T} \, {}^{G}\vec{v}'_{olc_T} + \frac{1}{2} m_U \, {}^{G}v_{olc_U} \, {}^{G}v'_{olc_U} + \frac{1}{2} m_F \, {}^{G}\vec{v}_{olc_F} \, {}^{G}\vec{v}'_{olc_F}$$

$$+ \frac{1}{2} m_H \, {}^{G}\vec{v}_{olc_H} \, {}^{G}\vec{v}'_{olc_H} \quad \textbf{(4-55)}$$

$$+ \frac{1}{2} \vec{\omega}_T \boldsymbol{I}_T \vec{\omega}'_T + \frac{1}{2} \vec{\omega}_U \boldsymbol{I}_U \vec{\omega}'_U + \frac{1}{2} \vec{\omega}_F \boldsymbol{I}_F \vec{\omega}'_F + \frac{1}{2} \vec{\omega}_H \boldsymbol{I}_H \vec{\omega}'_H$$

$$PE = m_T g \, {}^{G}\vec{x}_{olc_T}(3) + m_S g \, {}^{G}\vec{x}_{olc_U}(3) + m_E g \, {}^{G}\vec{x}_{olc_F}(3) + m_W g \, {}^{G}\vec{x}_{olc_H}(3) \quad \textbf{(4-56)}$$

where $m_i \; i\epsilon[T, U, F, and \; H]$ is the mass of the torso, upper arm, forearm, and hand. ${}^{G}\vec{x}_{olc_i} \; i\epsilon[T, U, F, and \; H]$ is the location of the center of mass of each segment from the global origin located in the middle of the torso. ${}^{G}\vec{v}_{olc_i} \; i\epsilon[T, U, F, and \; H]$ is the velocity of the center of masses in the global frame.

## 4.5 Aim 1: Summary of Findings

With the completion of Aim 1, kinematic and kinetic based features were extracted from an IMU-based motion capture system with a performance that was comparable to the gold standard optical tracking system. This section presents the findings of the Aim 1 research, which led to the development of Aim 2 discussed in chapter 5.0.

### 4.5.1 Preliminary Feature Selection

---

*Measure of success: Define a set of kinematic and kinetic features to assess and predict stroke severity*

---

The literature review in chapter 3.0 described motion features that correlate with stroke severity but have not been implemented in a machine-learning algorithm to predict stroke severity for new patients. In this project, all motion features were selected using the following criteria: feature set had to (1) include all 10 joint angles mentioned in Table 1 (2) capture range of motion of each joint, and (3) capture the time based and/or range based relationship between joint motions.

For this aim, a subset of the motion features from the studies covered in the literature review was chosen to initialize the list of features. For example, the peak angular velocity was measured at each joint angle due to the results shown in Murphy's work (Murphy M.A., 2011). Features were also chosen by analyzing kinematic and kinetic plots between healthy and stroke patients within the studies. The goal was to find features that can distinguish healthy patients from stroke patients. Figure 20 shows an example of this comparison between patients.

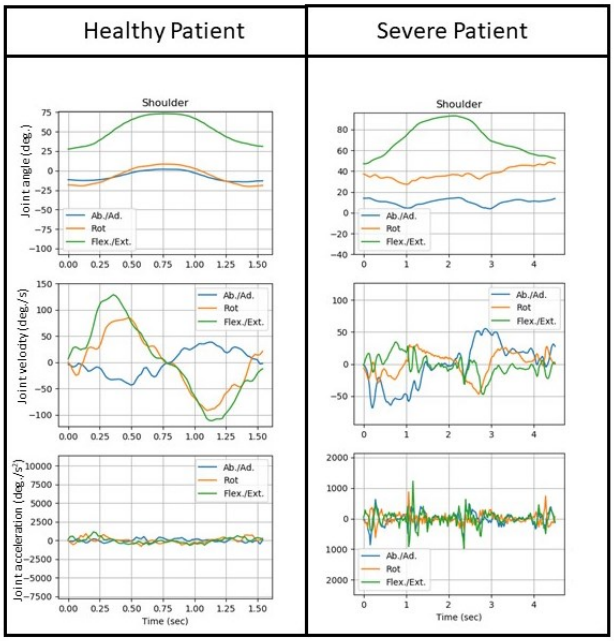



**Figure 20: Side by side comparison of shoulder kinematics for a healthy and stroke patient during the pointing task for Target 9 (Center target).**

From this plot, an observation was made that the shoulder-joint motion of a healthy patient had one distinct peak and valley within its velocity profile, while the stroke patient's velocity profile was more sporadic. This led to selection of the three features: amplitude of the dominant frequency, the number of zero crossings, and variance of each joint velocity. These features were later defined for all 10 joint angles. Examples of these motion features, extracted from the shoulder flexion joint velocity, are shown in Figure 21 – Figure 26.

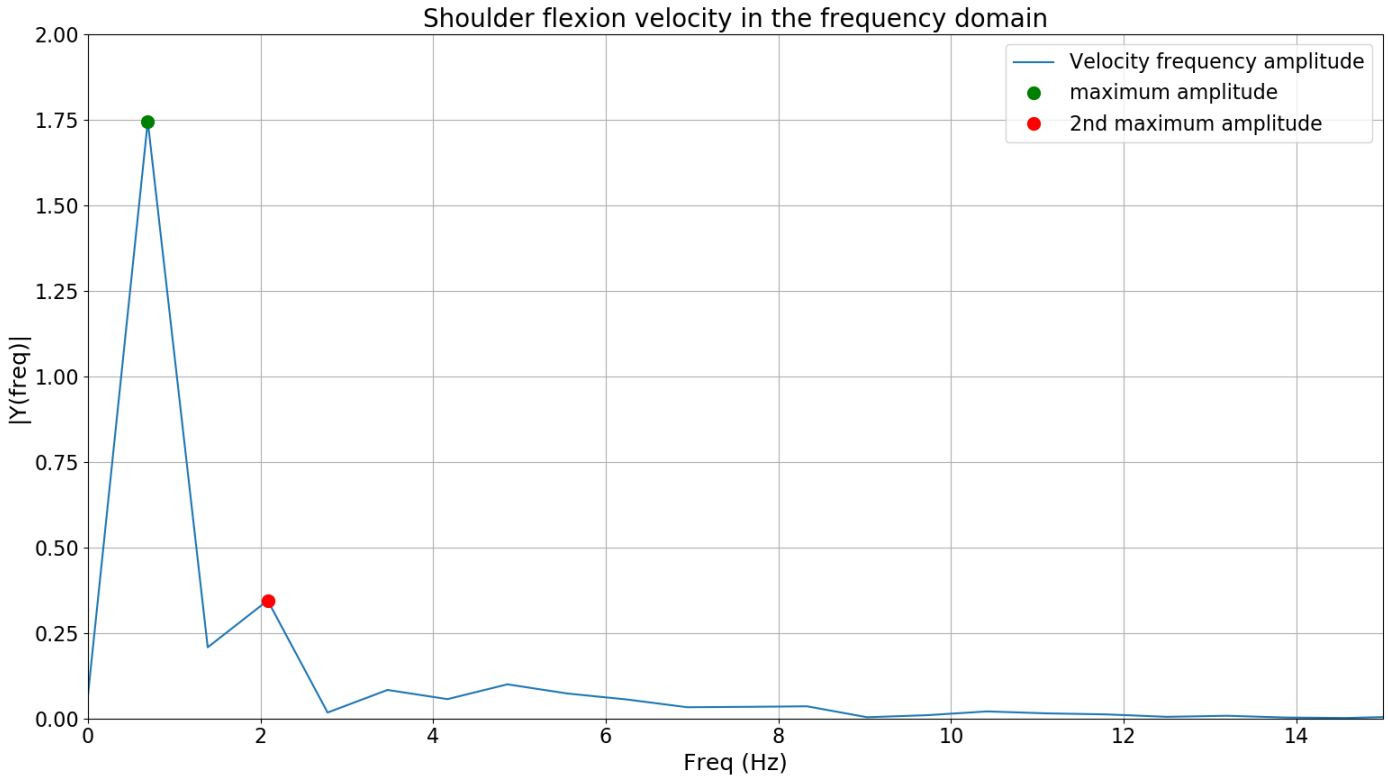



**Figure 21: Shoulder joint velocity of a healthy patient in the frequency domain with the two dominant frequency amplitudes highlighted in green and red.**

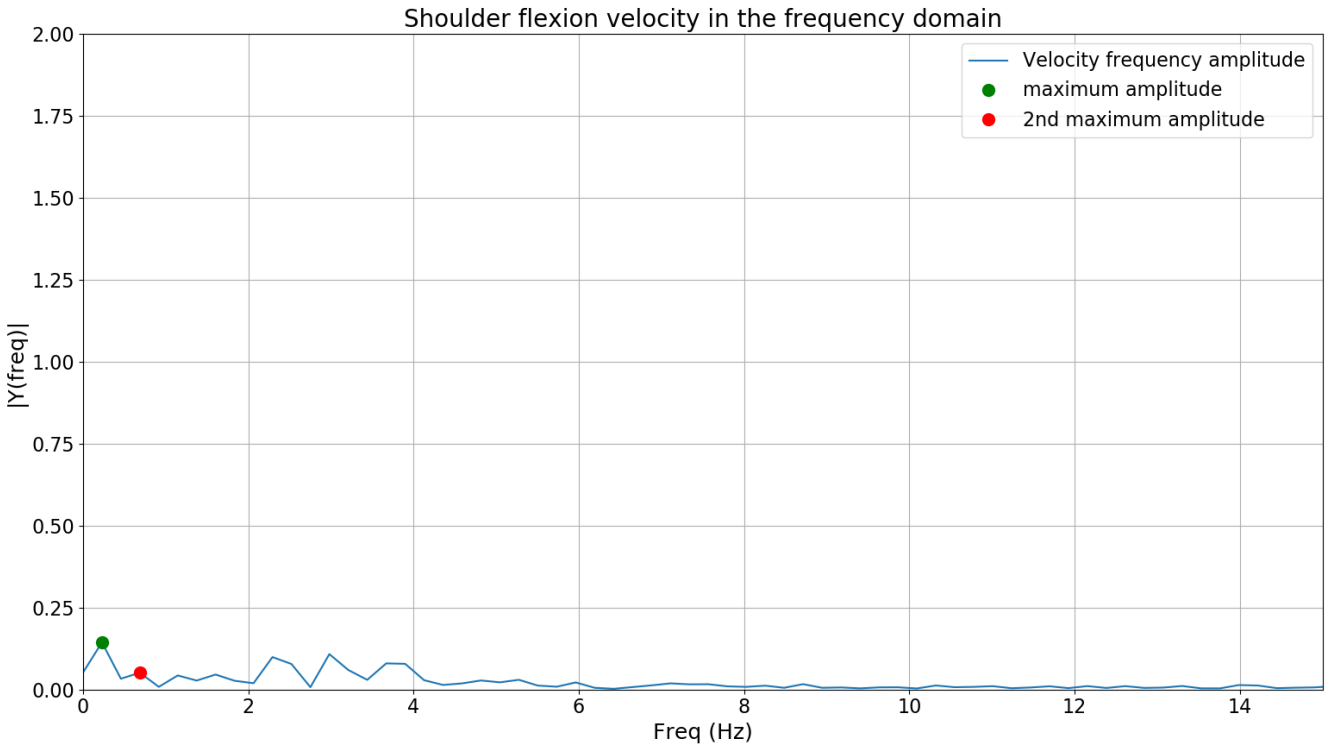



**Figure 22: Shoulder joint velocity of a severe stroke patient in the frequency domain with the two dominant frequency amplitudes highlighted in green and red.**


**Figure 23: Shoulder joint velocity of a healthy patient with the mean and the +/- standard deviation away from the mean shown in the solid and dotted black lines.**

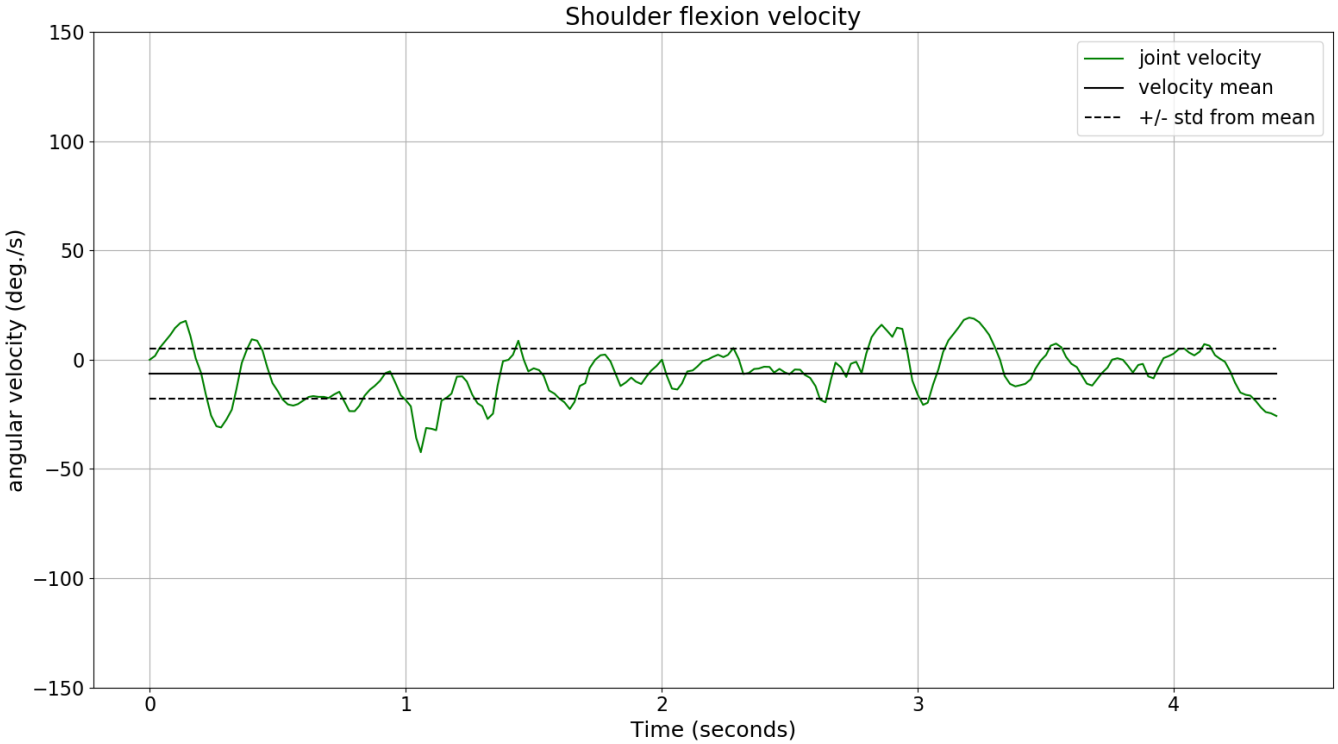



**Figure 24: Shoulder joint velocity of a severe stroke patient with the mean and the +/- standard deviation away from the mean shown in the solid and dotted black lines.**

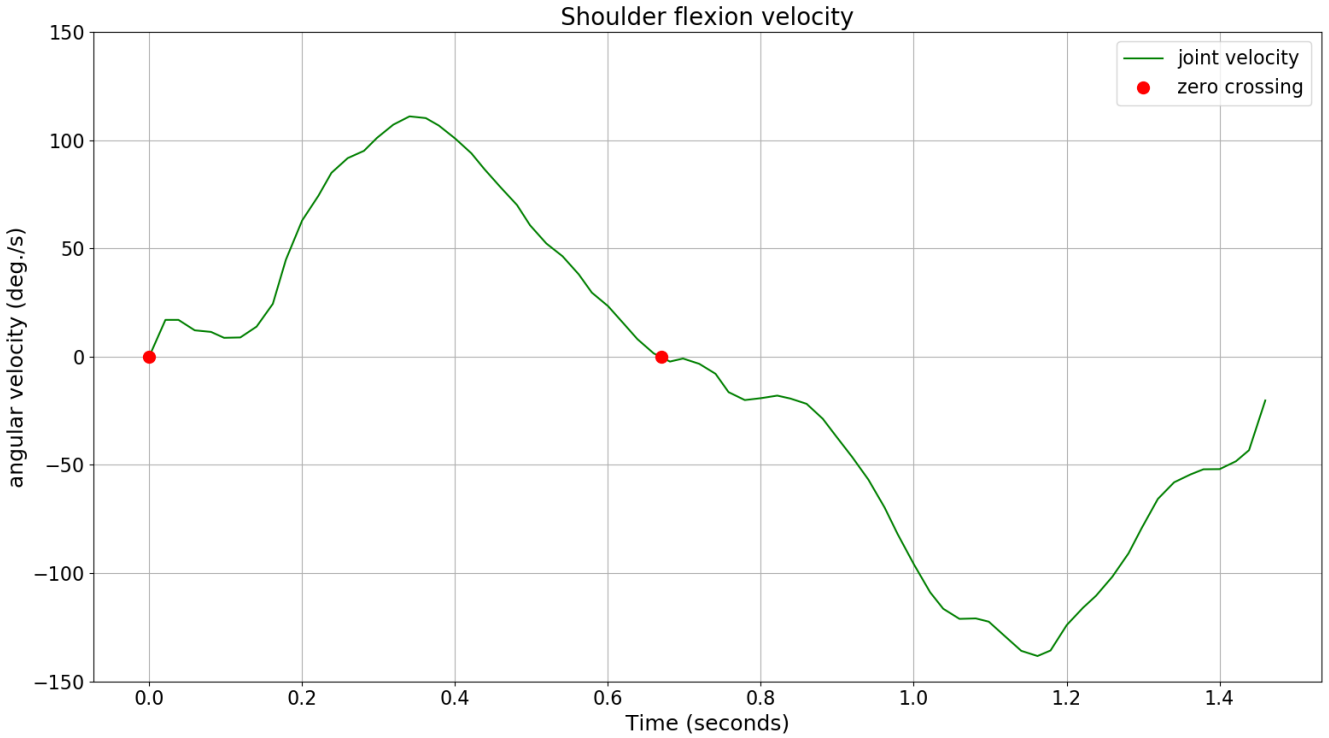



**Figure 25: Shoulder joint velocity of a healthy patient with the zero crossings (when the velocity reaches zero) highlighted in red.**

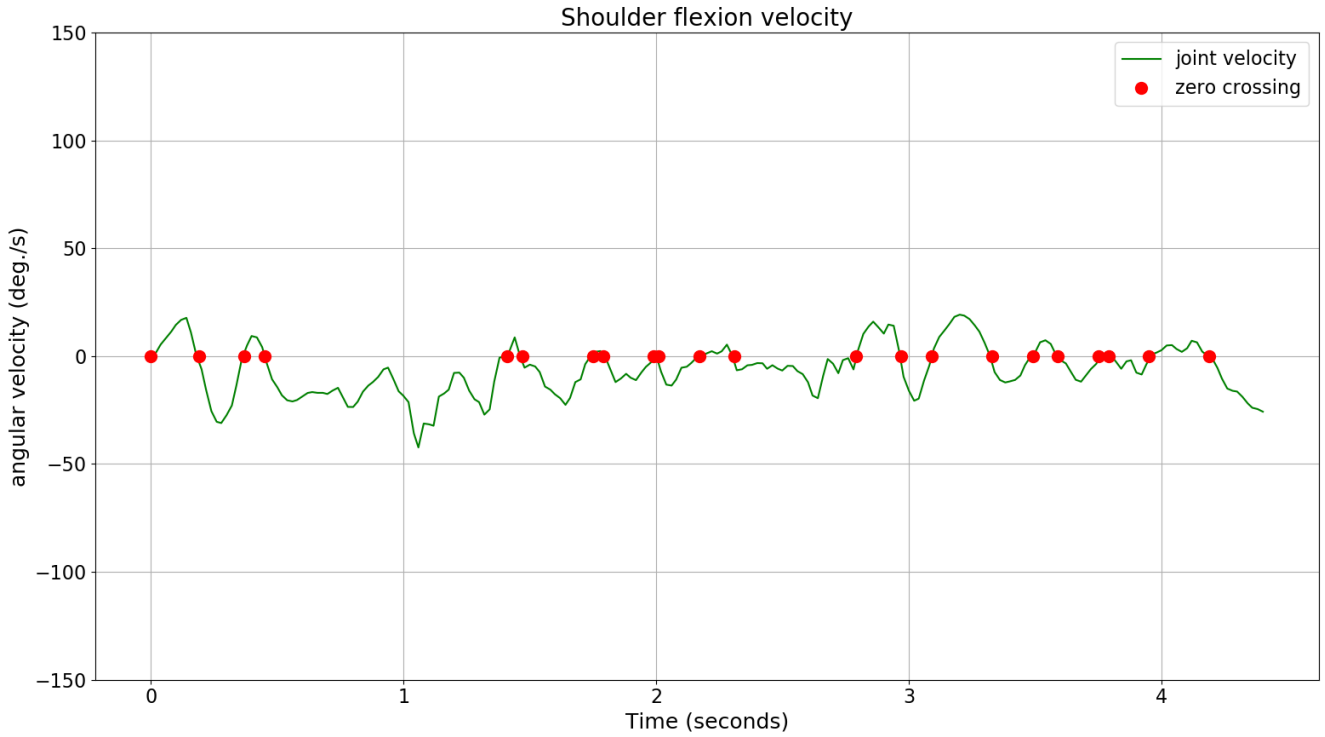



**Figure 26: Shoulder joint velocity of a severe stroke with the zero crossings (when the velocity reaches zero) highlighted in red.**

Within each motion feature demonstrated in Figure 21 – Figure 26, there is a clear difference in the motion patterns between a healthy and stroke patient. The same heuristic approach was repeated to explore other motion features within the feature set.

It is important to note that the feature selection was an iterative process. Aim 1 was revisited throughout the study if the measure of success for Aim 2 was not fulfilled (see section 1.1 for the measure of success of Aim 2). Once the feature set was defined, it was used as an input for the SVM to predict stroke severity in the patient population. If the F1 score (explained later in section 5.4) was not between 0.95 – 1, new features were defined and extracted from the IMU readings following the procedures outlined in Aim 1. Then the SVM used the feature set with the new features included to predict the stroke severity again and this iterative process was repeated until the highest F1 score was obtained. The final feature set that was selected for classification in chapter 5.0 is shown in Appendix B.

### 4.5.2 Feature Calculation

---

*Calculate motion features from the raw data that classify stroke severity.*

---

All the features defined in the first measure of success were calculated once the joint kinematics were collected from the IMUs. For the kinematic based features, statistical or frequency-based calculations were applied to get a scalar value that summarizes the patient's kinematics throughout the trial. These calculations were all done in Python, where the frequency-

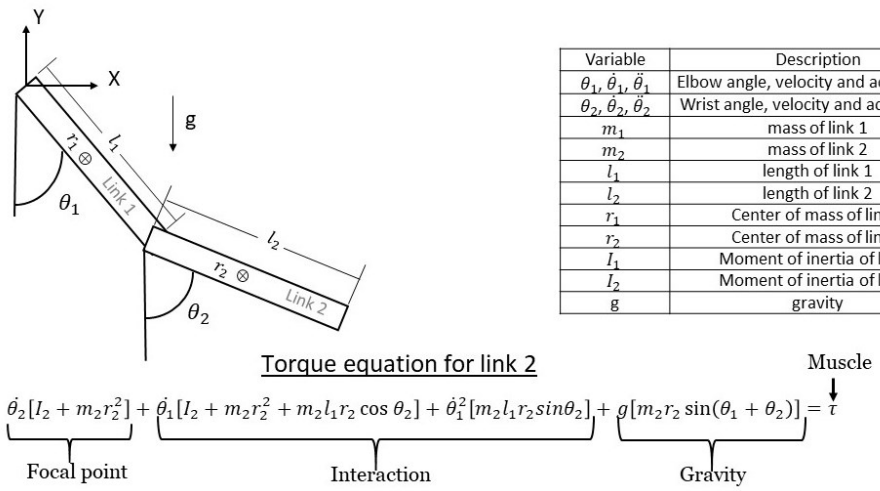
based calculations used the Scipy library to perform a discrete Fourier transformation (DFT) to represent the angular velocity and acceleration in the frequency domain. An example of how the amplitude of the dominant frequency was calculated for the joint velocities is shown in equations (4-57) and (4-58).

$$\dot{\Theta}_k = \sum_{n=0}^{N-1} \dot{\theta}_n \cdot e^{-\frac{i2\pi}{N}kn} \tag{4-57}$$

$$f_{max} = \max \dot{\Theta} \tag{4-58}$$

where $\dot{\Theta}_k$ represents the joint velocity $\dot{\theta}_n$ in the frequency domain at time step n. Equation (4-47) demonstrates the amplitude of the dominant frequency calculated from the joint velocity in the frequency domain.

The kinetic features were defined by calculating the equation of motion using the method described in section 4.4. The torque was then split into the focal point, gravity, interaction and muscle torque terms as described in (Raj S., 2018). An example of how this can be done for a simple planar double pendulum system is shown in Figure 27.




| Variable | Description |
|---|---|
| $\theta_1, \dot{\theta}_1, \ddot{\theta}_1$ | Elbow angle, velocity and acceleration |
| $\theta_2, \dot{\theta}_2, \ddot{\theta}_2$ | Wrist angle, velocity and acceleration |
| $m_1$ | mass of link 1 |
| $m_2$ | mass of link 2 |
| $l_1$ | length of link 1 |
| $l_2$ | length of link 2 |
| $r_1$ | Center of mass of link 1 |
| $r_2$ | Center of mass of link 2 |
| $I_1$ | Moment of inertia of link 1 |
| $I_2$ | Moment of inertia of link 2 |
| g | gravity |

**Figure 27: An example of defining the focal point, interaction, gravity and muscle torques for link 2 in the double pendulum.**

For this dissertation, the torque terms were extended to the 10 DoF equations of motion used to model the torque for the upper extremities. Since the equation is highly nonlinear, the terms were not as easy to identify as the example above. Matlab (Mathworks ®) Symbolic Math Toolbox was used to calculate the equations of motion from the Lagrange equation defined in equations (4-52) – (4-54) and to identify the focal point, gravity, interaction and, muscle torque terms. The equations were used in an inverse dynamics approach, where the joint kinematics from the IMU were used as the input to the equation of motion to get the joint torques. Once the joint torques were calculated, they were converted into scalar values to use as motion features using the same method mentioned for the joint kinematics.

### 4.5.3 IMU Validation

---

*Measure of success: IMU kinematic validation [Desired outcome: normalized cross-correlation 0.8 – 1].*

---

To validate the accuracy of the IMU system, the joint angles calculated through the IMUs were compared to the joint angles from the Optitrack system. In this validation study, the joint velocities and accelerations were ignored since the Optitrack takes the derivates of the angle adding noise to the signals which can amplify the error for the IMU.

The performance metric used in this study was the cross-correlation, which was chosen since there was an unknown lag between the signals from each system. The lag was most likely caused by the Optitrack directly measuring the voltage change in the push button (shown in Figure 5) while the IMU system used the wireless dongle to monitor the voltage. The standard equation for cross-correlation is shown in equation (4-59).

$$R_{12}(\tau) = \sum_{n=1}^{N} q_{IMU}[n] q_{Opti}[n-\tau] \quad \textbf{(4-59)}$$

where n is the time step, $q_{IMU}[n]$ is the joint angle from the IMU and at time step n, $N$ is the length of the IMU joint signal $q_{Opti}[n-\tau]$ is the Optitrack joint angle delayed by $\tau$. To normalize the output of the cross-correlation between [-1,1], equation (4-59) was modified as shown in equation (4-60).

$$R_{12}(\tau) = \frac{1}{N}\sum_{n=1}^{N}\frac{(q_{IMU}[n] - \bar{q}_{IMU})(q_{Opti}[n-\tau] - \bar{q}_{Opti})}{\sigma_{q_{IMU}}\sigma_{q_{Opti}}} \quad \textbf{(4-60)}$$

where $\sigma_{q_{IMU}}$ and $\sigma_{q_{Opti}}$ are the standard deviation of IMU and Optitrack angle signals respectively. The goal was to determine the best cross correlation through adjusting the lag between the two signals through $\boldsymbol{\tau}$. Since the laptop running the IMU data logging experienced varying delays between trials, the optimal $\boldsymbol{\tau}$ had to be found for each trial evaluated. All ten joint angles were observed to measure the accuracy of the system. The results for one of the trials are shown in Figure 28.

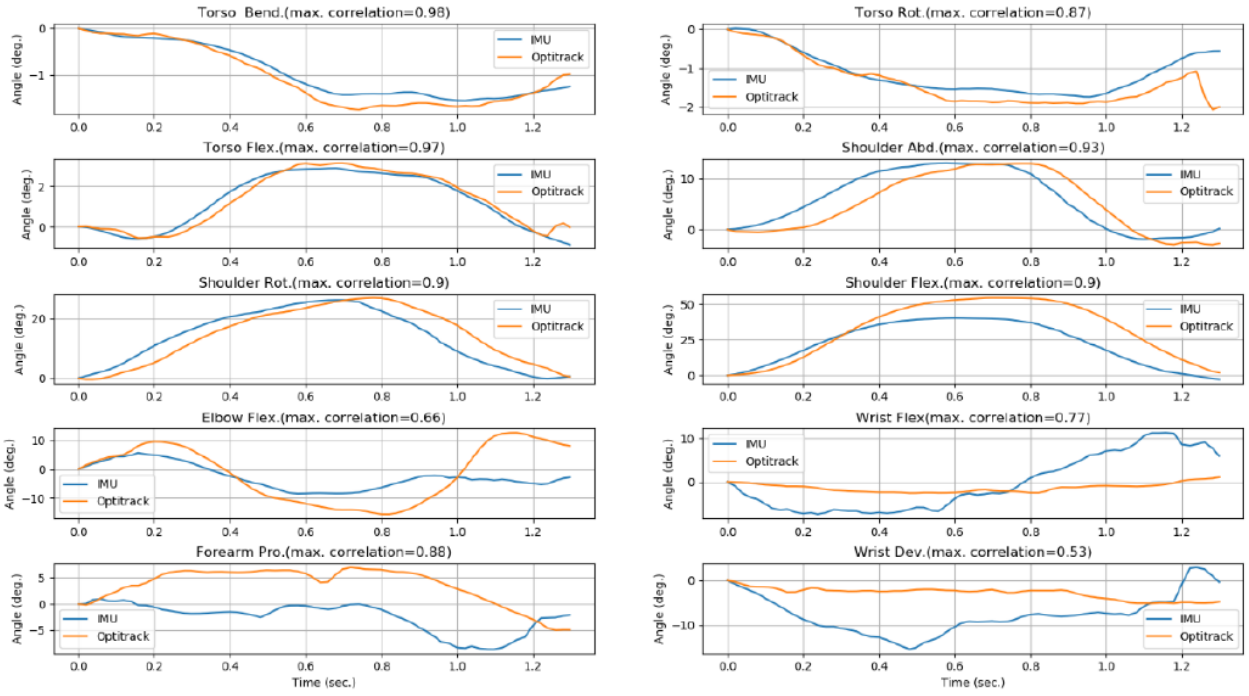



**Figure 28: IMU and Optitrack comparison with maximum cross correlation for a 40% arm length pointing to target 9 trial.**

The results in Figure 28 show that the IMU system correlated well (meeting the measure of success of 0.8 or above) with the torso and shoulder joints in all axes. The performance in the other joints deteriorated with a correlation between about 0.5 – 0.66. To see if this trend in IMU performance was consistent throughout the trials, 4 trials were randomly selected for each task and the normalized cross-correlation was averaged across the 4 trials for each joint. The result for each type of target is shown in Figure 29 – Figure 31.

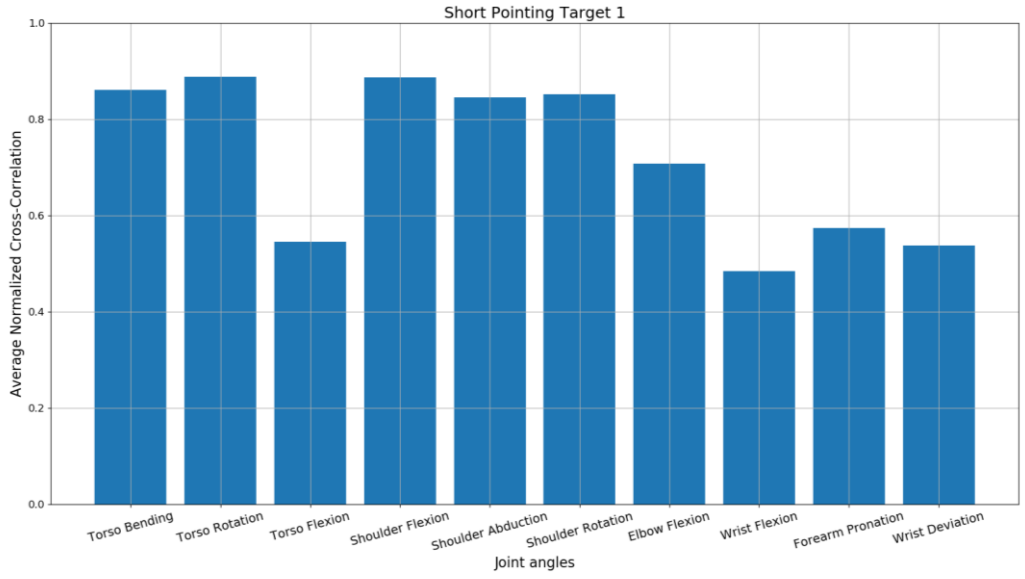



**Figure 29: Average normalized cross correlation over 4 trials for the short pointing target 1 task.**

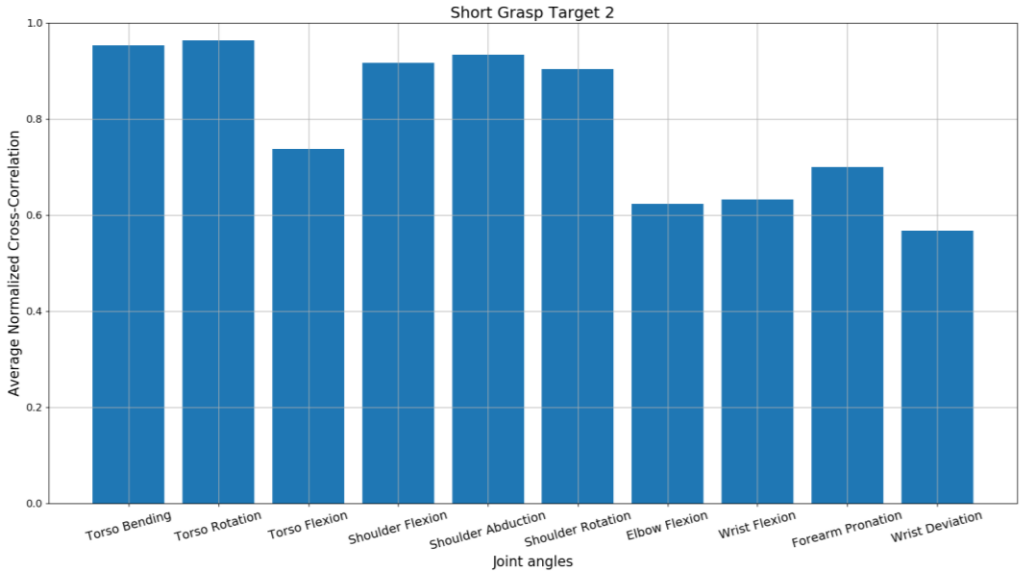



**Figure 30: Average normalized cross correlation over 4 trials for the short grasp target 2 task.**

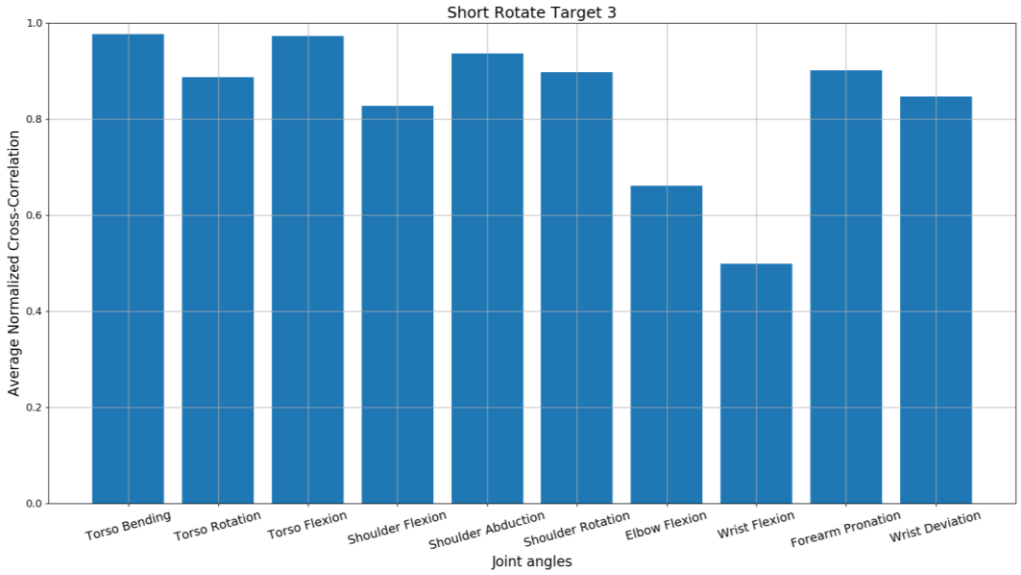



**Figure 31: Average normalized cross correlation over 4 trials for the short rotation target 3 task.**

The results in Figure 29 – Figure 31, show that the torso flexion, wrist flexion forearm pronation, and wrist deviation had low cross-correlation values for the short pointing and grasp tasks. While for the short rotation task, the forearm pronation and wrist deviation saw an improvement in their performance with cross-correlation values of 0.9 and 0.85, respectively. It is assumed that the lower performance at the torso flexion, wrist flexion, forearm pronation, and wrist deviation joints was caused by the misalignment of the body coordinate systems defined for each motion capture system. The IMUs assume that the body Y-axis (longitudinal axis along segment) is parallel with gravity, while the Optitrack body Y-axis was defined as the two points connecting the proximal and distal joints surrounding each segment.

The readings from the gyroscope also factor into the misalignment. If participants had any off-axis motion as they were flexing a given joint during the alignment calibration process, it would cause both the body X and Z axes to not point towards their true anatomical directions. The calibration motions were more difficult to perform for stroke participants. For severe stroke participants, students helping with the research would support the participant throughout the calibration step. This would help to ensure proper motions, but it was hard to guarantee the same alignment accuracy as with a healthy participant.

Although the IMU system did not correlate well with the Optitrack system in some scenarios, it did not deviate enough where the overall characteristic of the motion cannot be recognized (The results for all of the tasks is shown in Appendix D). Furthermore, the joint angles extracted from the IMUs were useful in developing the classification and personalized rehabilitation process. The measure of success was consistently met for the torso and shoulder angles (i.e. torso bending, torso bending, shoulder flexion, shoulder abduction, and shoulder rotation) while the performance of the torso flexion, elbow flexion, forearm pronation and wrist angles (wrist flexion and wrist deviation) varied between trials and tasks. Although there were some cases of disagreement between the two systems, it was determined that there was sufficient agreement in important measures for the IMU data to be used in developing a method with which to classify stroke severity and later to rank task difficulty. As can be seen later in chapter 5.0, the joint angles that showed poor performance (normalized cross-correlation lower than 0.8) were rarely chosen as a motion feature to predict stroke severity, and subsequently to rank tasks. To fully incorporate all motion features in the future, more work can be done to further close the gap between IMU and Optitrack results, with specific emphasis on wrist and elbow motions.

## 4.6 Aim 1: Measure of Success Achievements

With the completion of Aim 1, the measures of success for selecting and calculating a feature set (sections 4.5.1 and 4.5.2) was met with the creation of the feature set used as the input to the machine learning algorithm mentioned in Aim 2 (chapter 5.0). As mentioned before, the feature selection and calculation steps were revisited if the F1 score (part of Aim 2) was not between 0.95 – 1.

As for the IMU Validation step (section 4.5.3), the measure to have the cross-correlation of the joint angles for the IMU and Optitrack to be 0.8 – 1 for all joint angles was met consistently for the torso and shoulder joints but not always met for the elbow and wrist joints. The IMU results were able to characterize the reaching motion well enough to continue to Aim 2 but future adjustments in the body coordinate system definitions for the IMU and Optitrack systems should be made to improve the joint angle agreement in future studies. These adjustments should include how to have the motion-based alignment matrix process of the IMUs agree with the bony landmark digitization process of the Optitrack system. Otherwise validating that the remaining IMU results are satisfactory for stroke severity classification, even without a good match to Optitrack, is an alternative outcome.

# 5.0 Aim 2: Use Machine Learning to Predict Stroke Severity

This chapter discusses how the motion features in chapter 4.0 were used to classify stroke severity. Due to the large feature set defined earlier, feature selection methods needed to be implemented to choose the most useful features for classification. A machine-learning algorithm was applied to the data obtained from each task described in chapter 3.0. Participants were classified into 3 classes that consisted of healthy, mild stroke and moderate/severe stroke. The results showed that a larger pool of participants is needed to justify that the top features are appropriate, but the machine learning process proves to be impactful for future studies.

Section 5.1describes the rationale for using machine learning to classify stroke severity and the theory behind the algorithms used. Section 5.2 provides an overview of the support vector machine algorithm and its application in classifying stroke severity. Section 5.3 introduces the Synthetic Minority Oversampling Technique (SMOTE) that was used to interpolated classes that had fewer sample points than necessary to effectively run the algorithm due to a participant's inability to complete a task. Section 5.4 gives an overview of how the F1 score was used as a performance metric for the classifier. Section 5.5 describes the feature selection process to find an optimal subset of features from the features described in section 3.2. The application of the feature selection process and the results is shown in section 5.6. Finally, section 5.7 summarizes why the measure of success was not met and methods for future studies to adopt in order to meet the Aim 2.

## 5.1 Theory and Methods

Given the large set of motion features defined in chapter 3.0 and obtained in chapter 4.0, a method needs to be developed that can correlate these features with the appropriate FMUE score. A function that maps the FMUE score and the motion features will help create an automated system that can diagnose stroke severity and compliment the feedback provided by a clinician. Once this function is defined, it will need to be able to calculate future FMUE scores from patients outside of this study using their motion features. This will provide a diagnostic tool for a stroke patient outside of the clinic. This aim falls under the category of predictive analysis since the objective is to use previously collected data to predict future outcomes. The difficulty in accomplishing this task lies in the volume of data and the complexity of relationships between features and the FMUE scores.

In the ideal situation, the dimensions of the feature set can be reduced to one or two important features and a simple function can be created to map them to the FMUE score. In most real-world applications, this is not a simple task and requires more sophisticated tools. Luckily, this is a common problem seen in most predictive analysis applications and a vast number of algorithms have been developed for this purpose.

Statistical methods have been implemented for centuries to find relationships between variables in complex processes by using a set of observations over time. The two common forms of statistical models are linear regression and analysis of variance (ANOVA). Lee (Lee T.H., 1999) used a multiple regression model to find a relationship between 22 gait variables (retrieved from gyroscopes and force plates) and target variables (age, velocity, weight, and height) to find relevant motion features that show the effects of total knee arthroplasty operation. Some examples of the gait variables are cadence, step length gait cycle and step width. The multi regression results

showed that velocity had the most significant effect on the gait variables followed by weight, age, and height.

With the use of statistical models to achieve a better understanding of a given system, statisticians have tried to extend these same models for the application of predictability. While others sought to extend these statistical tools into what is known today as machine learning.

During the 1950s and 1960s, machine learning was built upon the principles of statistics as computers started to become more powerful (Kononenko I., 2001). Even today, machine learning is commonly confused with statistical models and they are asserted by some to be one and the same. Breiman (Breiman L., 2001) highlights the difference between these two approaches. He states that in the data modeling culture (that is, among researchers who use statistical models), the assumption is made that nature is a stochastic process that can be modeled using methods such as linear or logistic regression. While within the algorithmic modeling culture (researchers that use machine learning), it is assumed that nature is too complex to model and a better approach is to focus on modeling the predictability of action. An illustration of these two assumptions is shown in Figure 32.

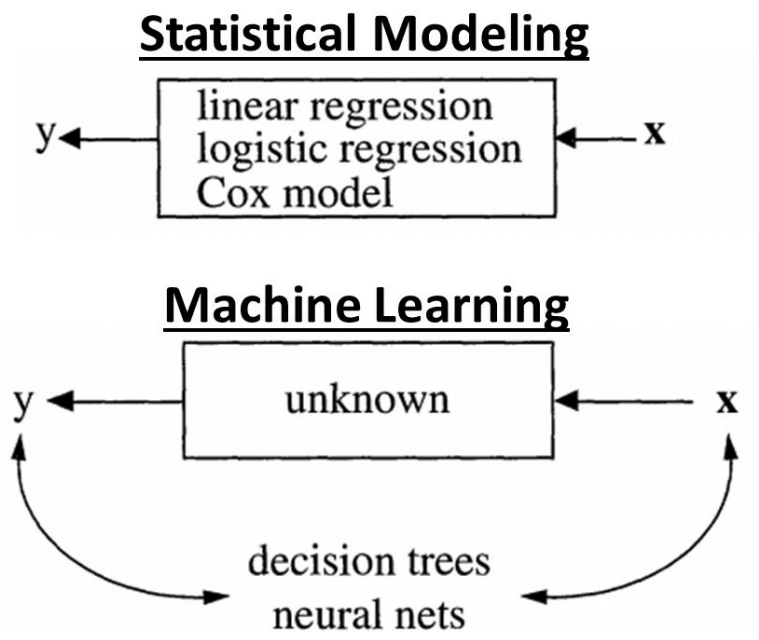



**Figure 32: (Breiman L., 2001) shows the two different assumptions made for modelling predictability of an action.**

Through his own experience, Breiman learned how misleading regression models and the r-square values (a measure of goodness-of-fit) can be. Most early studies concluded that a given model was accurate based on the r-square value and did not further test whether that model could generate estimations of future data. Bickel (Bickel P.J., 2006) stated that goodness-of-fit tests have trouble rejecting the hypothesis of a well-fit curve unless there is a large disparity between the data and the model. Through observing these past studies, statistical models have proven to give a nice estimate of relationships between variables but should not be used for predicting future outcomes.

By viewing the actual relationship between the input (x) and output (y) as a black box, the combination of computational power and statistics helps to find another relationship between variables. This, in turn, helps to strengthen the predictability of the machine learning model. Machine learning models have a lot in common with statistical models, but machine learning derives methods from statistics, probability theory, computational complexity theory, control theory and many other fields (Meyfroidt G., 2009). This causes the models to become more complex and makes it harder for researchers to interpret the model's equations.

Fast forward to today, the advancement of computers has also provided the ability to store large datasets to strengthen the application of machine learning. This allows the reduction of bias in the models which makes it more robust to predicting outcomes for various cases. Algorithms such as k-nearest neighbors, naïve and semi naïve Bayes, neural networks and decision trees have been used for medical diagnostics (Kononenko I., 2001). In an earlier study with Kononenko, Kukar (Kukar M., 1996) used semi naïve Bayes for the prognostics of femoral neck fracture recovery.

Outside of using machine learning for diagnostic predictions with data stored on medical devices provided on computers within the hospital, this work has been extended to motion capture technologies to observe motor skills, which allows data to be collected remotely and untethered to the hospital or clinic. An engineering application of machine learning has been in the use of visual and inertia-based sensors for biometric recognition of an individual.

This chapter presents the framework for using a machine learning method, the support vector machine (SVM), to diagnose stroke severity when a large dataset is present. It is hypothesized that given enough data, a full scale of the FMUE score (0-66) can be mapped to kinematic and kinetic features and machine learning-based classification can be made at increasing

levels of resolution approaching that of the FMUE scale itself. Figure 33 shows a high-level flowchart of how stroke severity was classified. The remaining sections will discuss how this flowchart is implemented.

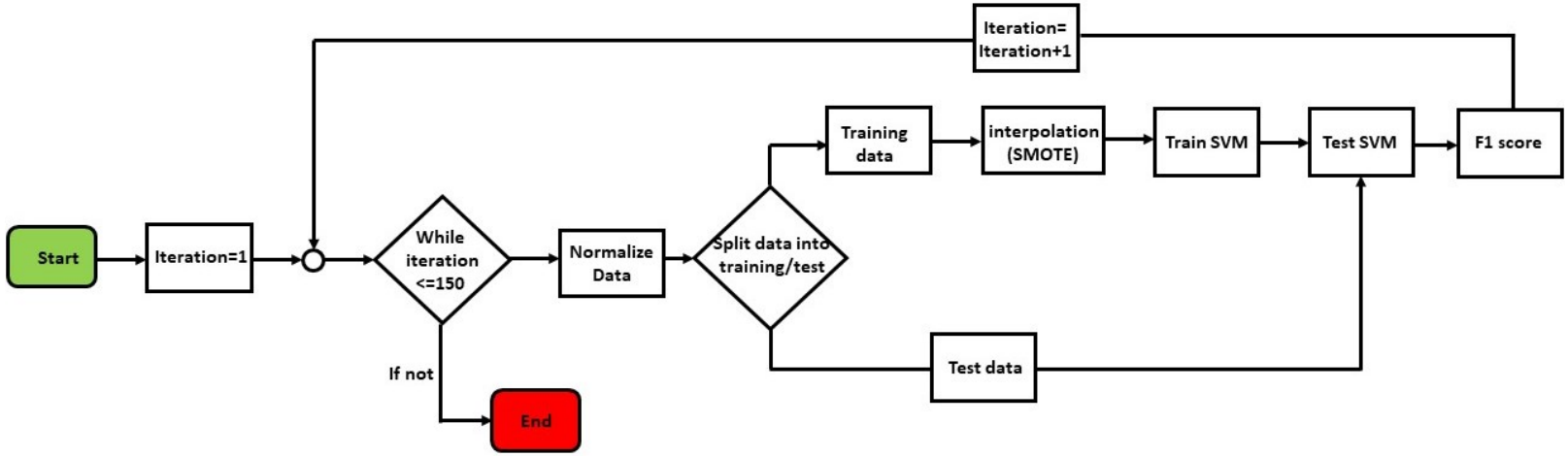



**Figure 33: Flowchart of how the machine learning algorithm was implemented to classify stroke severity.**

## 5.2 Stroke Severity Classification Using Support Vector Machines

To classify the stroke severity of the subjects in the study using IMU data, the features extracted from the data, as explained in chapter 4.0, were mapped to FMUE scores. The patients wore the IMUs and the features extracted from the inertial measurement data were used in a machine-learning algorithm to provide a classification that relates to an FMUE score. Independently, clinicians monitored the patients' motion to provide a reference FMUE, which was used to label data for training and testing of the machine learning algorithm. Then with this known mapping, future diagnoses can be made from the computer using machine learning algorithms.

After the features were extracted, each motion was represented in a multi-dimensional feature space. To differentiate between the skill sets that produced a motion, boundaries needed to be defined. An SVM is a supervised learning algorithm that creates optimal boundaries (or support vectors) between classes using labeled datasets. These boundaries are defined by decision rule functions that output the appropriate label given a feature vector. A linear decision rule function for a two-class problem is shown in equations (5-1) and (5-2).

$$(\vec{w} \cdot \vec{x}_i + b) \geq 1, \qquad if\ y_i = 1 \tag{5-1}$$

$$(\vec{w} \cdot \vec{x}_i + b) \leq -1, \qquad if\ y_i = -1 \tag{5-2}$$

$\vec{w}$, $\vec{x}_i$, $b$, and $y_i$ represent the weight vector, feature vector, bias and class label respectively for sample $i$. These constraints are then put into a Lagrangian function to find the optimal boundaries (or hyperplanes) between classes. This is shown in the equation (5-3).

$$L(\vec{w}, \alpha_i, \mathrm{b}) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1] \quad \textbf{(5-3)}$$

where $L$ is the Lagrangian function, $N$ is the total number of samples in the training dataset and $\alpha_i$ is the Lagrangian multiplier. The goal is to minimize $\vec{w}$, which will maximize the margin of the hyperplanes. An example of the classifier is shown in Figure 34.

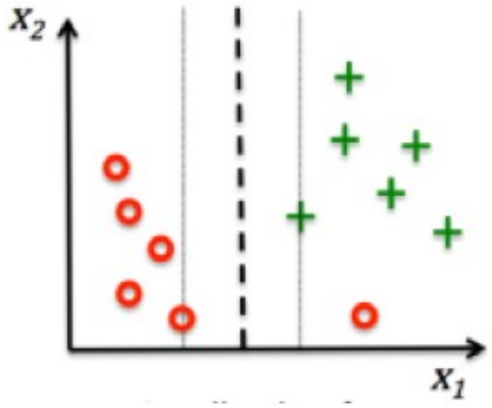



**Figure 34: Example of an SVM separation for two classes (Raschka S., 2016).**

Figure 34 represents a 2D (2 features) linear separation of two classes, represented by red circles and green plusses. The dashed line represents the hyperplane that the algorithm has found to

separate the classes (in this simple case, just a straight line). In some cases, the separation is not correct (e.g., the red circle on the right side of the dashed line), so the decision rule or the objective function that defines the hyperplane between the classes need to be modified. Modifications can involve creating a decision rule that generates complex nonlinear boundaries for a more optimal separation. With SVMs, the decision rule can be modified to transform the data into a space where other separation cases can be explored. The transformation functions that modify the decision rules are called kernels. The purpose of the kernel function is to transform variables into another feature space where nonlinear boundaries can be explored. In other words, it is mapping the linear decision rule to a nonlinear function. Multiple kernels have been designed to create complex boundaries such as polynomial, radial basis function (RBF) and sigmoid. Kernels can be inserted into the decision function as shown in equations (5-4) and (5-5).

$$\left(\vec{w} \cdot K(\vec{x}_i, \vec{x}_j) + b\right) \geq 1, \qquad if\ y_i = 1 \tag{5-4}$$

$$\left(\vec{w} \cdot K(\vec{x}_i, \vec{x}_j) + b\right) \leq -1, \qquad if\ y_i = -1 \tag{5-5}$$

where $K(\vec{x}_i, \vec{x}_j)$ can be one of the three,

$$K(\vec{x}_i, \vec{x}_j) = exp\left(-\gamma \left\|\vec{x}_i - \vec{x}_j\right\|^2\right), \text{RBF} \tag{5-6}$$

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d, \text{Polynomial} \tag{5-7}$$

$$K(\vec{x}_i, \vec{x}_j) = tanh\left(\gamma \vec{x}_i^T \cdot \vec{x}_j + \mathrm{r}\right), \text{Sigmoid} \tag{5-8}$$

$K(\vec{x}_i, \vec{x}_j)$ represents the kernel function and r and $\gamma$ are kernel parameters. Once a kernel function is inserted into the Lagrange multiplier equation, the new boundaries can be produced. A comparison of the kernel separations is shown in Figure 35.

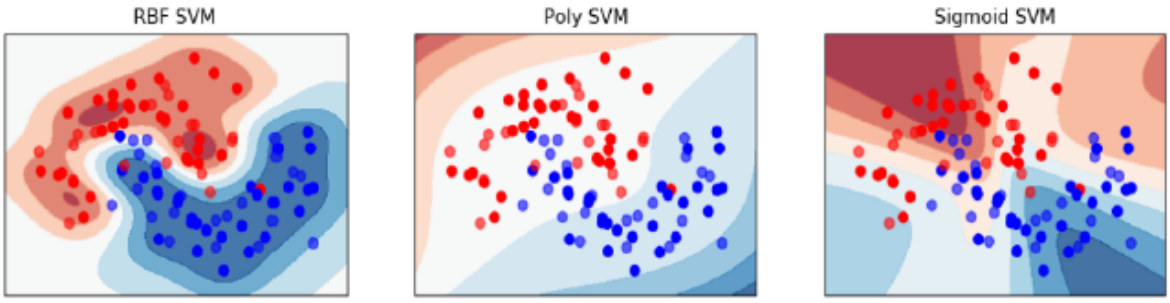



**Figure 35: Examples of different boundaries developed using various SVM kernel functions (Pedregosa. F., 2011).**

Along with picking the proper kernel that provides the optimal separation between classes, the characteristics of the hyperplanes can be further modified to fit the training data through tuning the model parameters. In order to make the orientation of hyperplanes less sensitive to outliers, slack variables are added to the constraints in equations (5-1) and (5-2) as shown in equations (5-9) – (5-11).

$$(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \varepsilon_i, \qquad if\ y_i = 1 \tag{5-9}$$

$$(\vec{w} \cdot \vec{x}_i + b) \leq -1 + \varepsilon_i, \qquad if\ y_i = -1 \tag{5-10}$$

$$\varepsilon_i \geq 0 \tag{5-11}$$

where $\varepsilon_i$ is the slack variable that adds a penalty to the objective function if $\vec{x}_i$ is an outlier. The series of slack variables are incorporated in the Lagrange multiplier in equation (5-12).

$$L(\vec{w}, \varepsilon_i, \alpha_i, r_i, b) = \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\varepsilon_i - \sum_{i=1}^{N}\alpha_i[y_i(\vec{w} \cdot \vec{x}_i + b) - 1] - \sum_{i=1}^{N} r_i\varepsilon_i \tag{5-12}$$

where *C* is the penalty parameter and $r_i$ is another Lagrange multiplier added to the equation. A small *C* lets the constraints be ignored which allows more outliers in the classification but creates a more generalized SVM model for future data, while a large *C* creates hard constraints that allow highly accurate classification but will overfit the data such that it may readily misclassify future datasets. It is up to the researcher to understand the data to find a balance between accuracy and model robustness.

For the nonlinear kernels, other parameters can be adjusted that affect the shape of the hyperplane. The polynomial kernel has the parameter, *d*, which changes the order of the polynomial function. A low order polynomial will create a smooth hyperplane while a high order polynomial creates a "bumper" hyperplane. For the RBF and sigmoid kernels, the γ seen in equations (5-7) and (5-8) can be tuned to determine the area of influence of points ($\vec{x}_i$) that affect the hyperplane. A smaller γ mainly focuses on far points from the hyperplane, while a larger γ mostly uses points close to the hyperplane. The effect of both *C* and γ is shown in Figure 36.

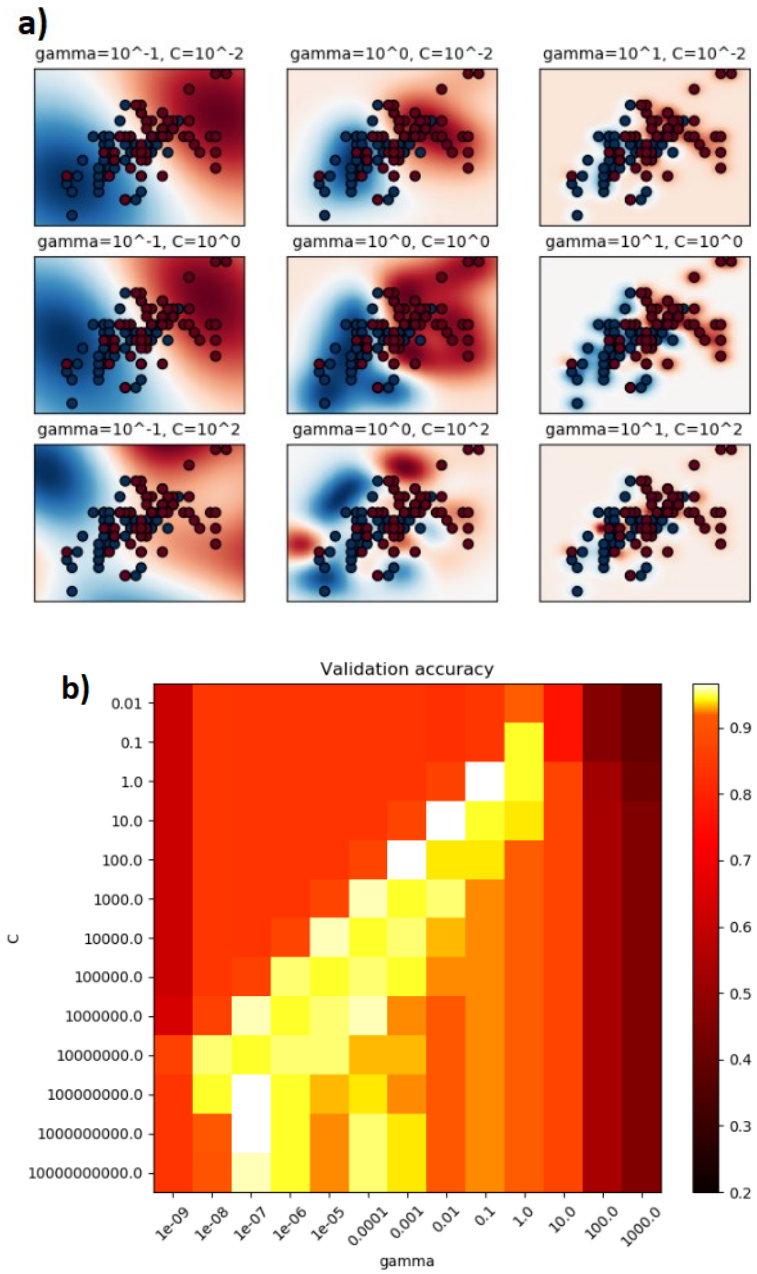



**Figure 36: a) Visual representation of tuning the C and gamma parameters for an RBF kernel SVM. b) A heatmap of the validaton accuracy with the given parameters. (Pedregosa. F., 2011).**

Figure 36 shows the boundaries (the white lines between the red and blue shaded areas) that classify the red and blue data points and the effects $C$ and $\gamma$ have on these boundaries. If classified correctly, the red points should only be surrounded by the red boundary and the blue boundary for blue points. As can be seen, a smaller $\gamma$ and $C$ has a smoother boundary but has a lot of misclassified points. A high $\gamma$ and $C$, on the other hand, obtains a high classification accuracy but fits tightly around each point. Figure 36b, shows a different interpretation of tuning the parameters by showing the SVM classification accuracy (where 1 or the white shade is 100% accuracy) with the heatmap of $\gamma$ vs $C$ parameters. As is shown in the figure, areas in the upper right (corresponding to $\gamma$ and $C$ of 1000 and 0.01 respectively) provide less accurate classification than areas in the lower left (where $\gamma$ and $C$ are 1e-7 and 1e8 respectively). To tune these parameters, a popular algorithm called the grid search method was implemented. The effectiveness of the model parameter tuning was measured by checking the model stability. This was done by running through $N$ iterations where the data is randomly split into 70% training and 30% testing data for each iteration. $N$ was set to the number of possible combinations of participant trials for that given task. If the model is stable, the classification accuracy will not fluctuate greatly between iterations.

The methods described above were all for a general two-class (binary) machine learning problem. For this dissertation, three classes were defined. The labels for the datasets were based on the FMUE score provided by a clinician monitoring the motion. The three classes were healthy, mild (FMUE≥47), and moderate/severe (FMUE<47). The two multi-class algorithms that can be used are the One-vs-Rest (OVR) and One-vs-One classifiers (OVO). OVR labels one class as +1 and the remaining classes as -1. The classifiers repeat this cycle until all classes are compared with the others. For the OVO classifier, $N(N\text{-}1)/2$ binary classifiers are defined, where $N$ is the number of classes (so in this case $N(N\text{-}1)/2 = 3$). In this method, each combination of classes will be

classified (healthy vs mild, healthy vs moderate/severe and mild vs moderate/severe). To classify a patient their data point is evaluated by all 3 classifiers and the class with the highest vote given is established as the final classification. Figure 37 show a graphical representation of OVR and OVO.

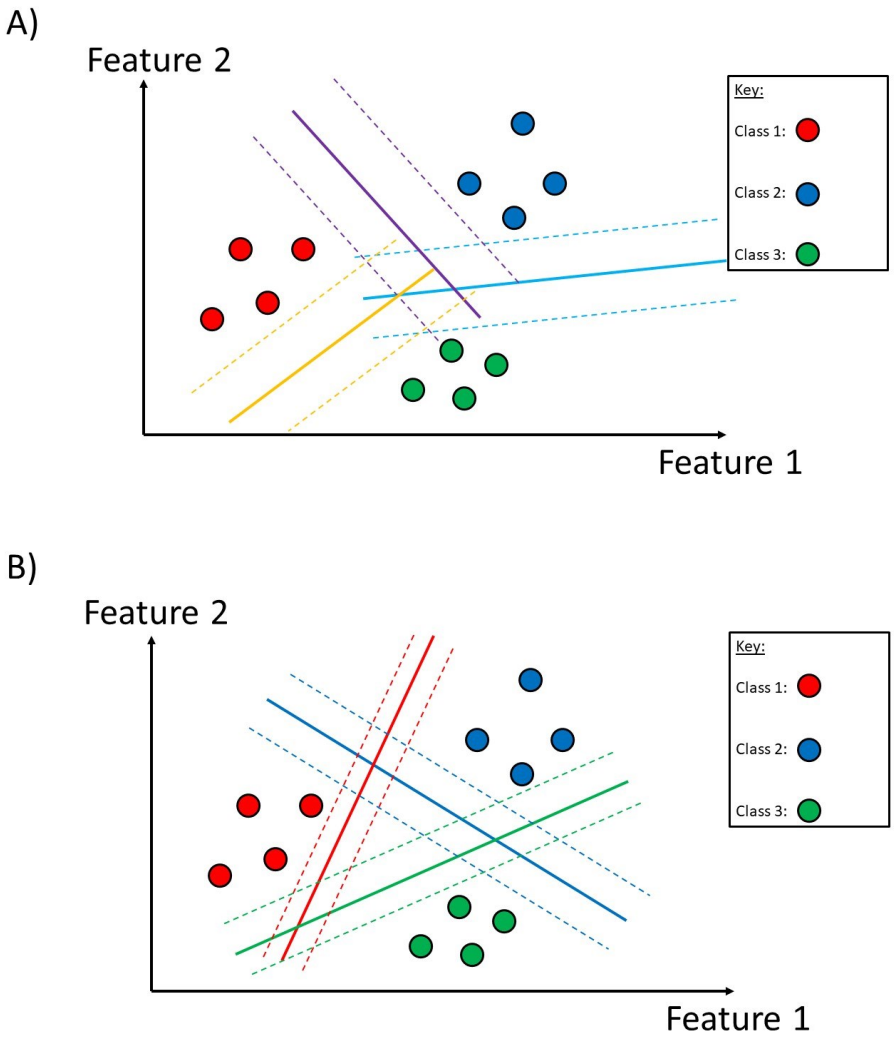



**Figure 37: a) Shows the OVR classification where optimal hyperplanes are created between a single class and the rest of the classes. b) Shows the OVO classification where a series of hyperplanes are created between each pair of classes.**

If the OVR classifier is used, there will be a large imbalance in the dataset to classify appropriately. For example, one iteration of the OVR will compare 10 healthy participants with the 20 participants that fall into the mild or moderate/severe categories. The classification that occurs will be extremely biased towards mild and moderate/severe participants. Since 10 participants were assigned to each class (healthy, mild, moderate/severe) the OVO classifier is a better multi-classifier candidate for this study.

### 5.3 Data Interpolation (Synthetic Minority Over-Sampling Technique)

Since each reaching task is a separate classification problem, there was some imbalance in the dataset for difficult tasks. This was due to some stroke participants not being able to complete the task during their session. To account for this imbalance the Syntenic Minority Over-Sampling Technique (SMOTE) was used to interpolate points in the minority class (the class with the least amount of sample points). The SMOTE process starts with a randomly selected data point of a participant (within the feature space) in the minority class and finding the $K$ nearest neighboring points within the class ($K$ was set to 2 for this study). In this case, other participants within the class that have similar features to the participant were selected. Then a new interpolated data point is defined calculating the distance between the two points and randomly multiplying it by a number between 0 and 1. Figure 38 shows graphically how this process is done.

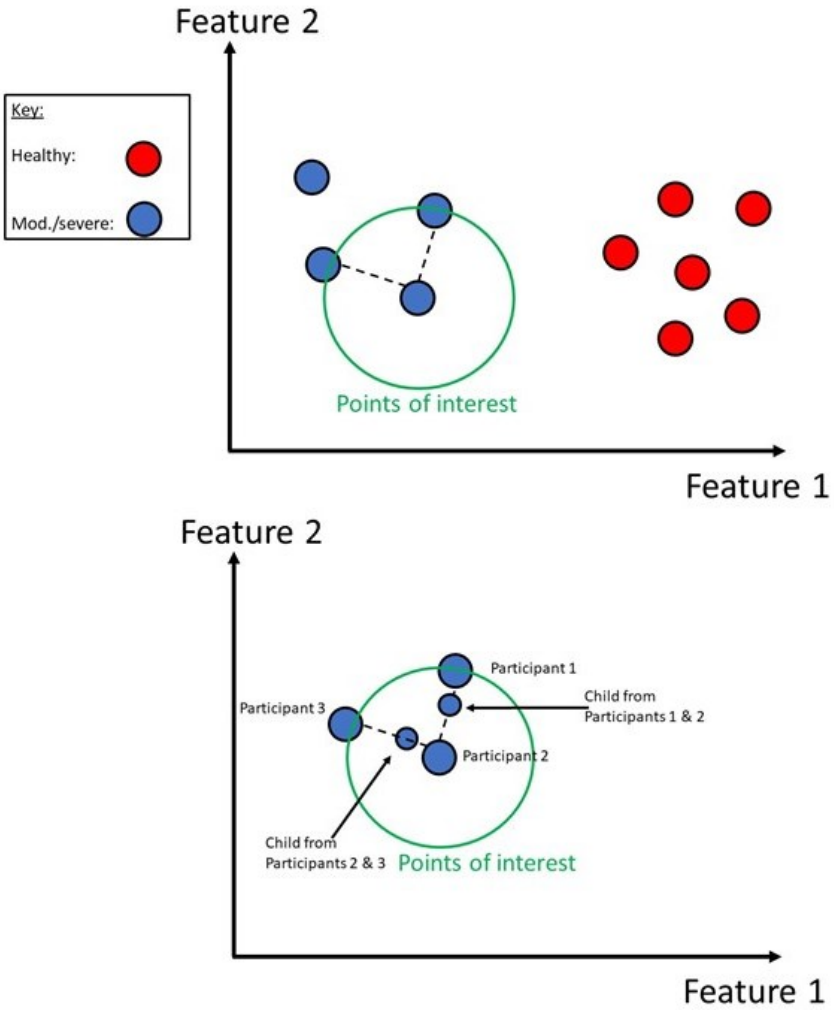



**Figure 38: Illustration of SMOTE being applied on the minority class.**

As shown in Figure 38, two participants are used to create a fake participant to test the algorithm. It is important to note that this method can only be applied to the training dataset. If it is also

applied to the testing data, it will cause a false sense of accuracy since the interpolated data point can have similar values as the training data from which it is derived.

Raziff (Raziff A.R.A., 2017) used this method to help classify participants based on their gait patterns using the accelerometer in their smartphones. Since the total participant population was 30 people and participants only recorded gait twice (once for the training set and the second time for the test set) the data was limited. This method helped to simulate a larger population to enable future studies in their machine learning algorithm.

Chawla (Chawla N.V., 2002) explored other applications of SMOTE for classification. This study incorporated SMOTE with naïve-Bayesian, RIPPER and decision tree (C4.5) classifiers with a variety of datasets with imbalanced data. The results show that SMOTE improved the classification accuracy when the minority classes were interpolated no more than the sample size of the majority classes.

Introducing this method limits the study from justifying the classification results as an exact replication of the general population. Just like Raziff, the results of this study were to justify the classification method for future studies to analyze a larger patient population.

### 5.4 Classification Performance Metric (F1 Score)

Although SMOTE reduces the disparities between the class sizes, it wasn't implemented to the point where the class sizes were even. SMOTE can be exhausted to the point where it just generates redundant information with similar interpolated points. To account for this imbalance the F1 score was used as a performance metric rather than accuracy (total correct classification divided by the total population size). If accuracy were selected, the score would be equally

weighted from all classes no matter the number of data points provided by that class. The F1 score uses both precision and recall in providing a classification score. Precision measures the consistency of the prediction while recall measures reliability (the number of true classifications that were classified correctly). This measure of the F1 performance metric is defined as the harmonic mean of precision and recall. The equations for precision, recall and F1 score are shown below.

$$\text{precision} = \frac{TP}{TP + FP} \tag{5-13}$$

$$\text{Recall} = \frac{TP}{FN + TP} \tag{5-14}$$

$$F1 = 2\left(\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}\right) \tag{5-15}$$

where *TP*, *FP* ad *FN* are true positive, false positive and false negative identified by the classifier respectively. For the 3-class case, all 3 F1 scores will be averaged to get the overall performance of the model. This average is expressed in equation (5-16).

$$\overline{F1} = \frac{F1^{Healthy} + F1^{Mild} + F1^{Mod./Sev.}}{3} \tag{5-16}$$

where $F1^{Healthy}$ , $F1^{Mild}$ $and$ $F1^{Mod./Sev.}$ are the F1 score for the one vs the one SVM models for healthy, mild and moderate/severe classes respectively. The kernel with the highest classification accuracy was selected for the final SVM algorithm. The goal is to have the SVM algorithm achieve an F1 score of 0.95 (A perfect score is equal to 1) on the *K*-fold cross-validation

(CV) test (*K* will equal 10). The SVM F1 score was tested by randomly selecting 70% of the data to train the SVM model and 30% to test the trained model. The F1 score was also implemented in the feature selection process to reduce the number of features needed to classify stroke severity. This is further explained in the next section.

## 5.5 Feature Selection

### 5.5.1 Feature Normalization

Before the features were used for machine learning algorithms, normalization techniques needed to be applied to the dataset. This is helpful especially if each feature has different units. Some of the most common normalization algorithms are the min-max, mean, and *z*-score normalization. The min-max and mean normalization techniques are normally applied to datasets where there are known hard limits.

In this dissertation, the *z*-score normalization was used to scale the data. Instead of normalizing the data with values [0,1] or [-1,1], the data is scaled to have a mean of zero and a standard deviation of 1. The calculation of the *z*-score is shown in equation (5-17).

$$z = \frac{x - \mu}{\sigma} \tag{5-17}$$

where $x$ is the feature, $\mu$ is the average of that feature and $\sigma$ is the standard deviation of that feature. The final *z*-score then represents the feature's distance from the mean by the factor of the standard deviation. An illustration of this distance is shown in Figure 39.

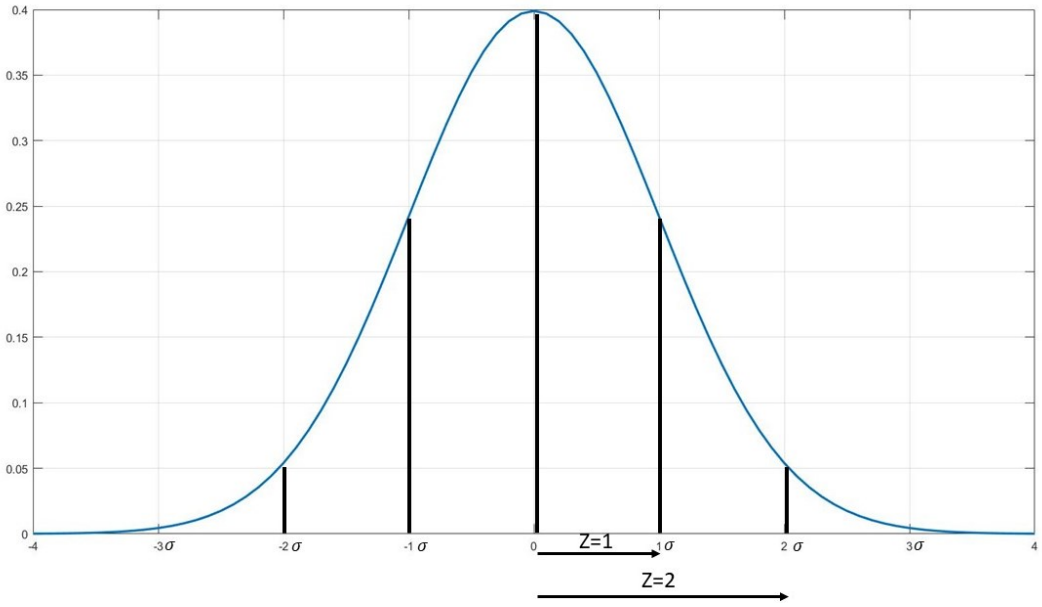



**Figure 39: Z-score represented on normal distribution curve.**

When the data is normalized with the z score, a data point with the z score equal to 1 would mean it is 1 standard deviation away from the mean.

To ensure the scaling factors are consistent with the training and test sets, the z score is implemented as shown in equations (5-18) – (5-19)

$$z_{train} = \frac{x_{train} - \mu_{train}}{\sigma_{train}} \quad \textbf{(5-18)}$$

$$z_{test} = \frac{x_{test} - \mu_{train}}{\sigma_{train}} \quad \textbf{(5-19)}$$

where $\mu_{train}$ and $\sigma_{train}$ are the mean and standard deviation calculated from solely the training data. These same parameters are then applied to the test data. Figure 40 shows a kernel density estimation of maximum amplitude of the shoulder flexion velocity signal in the frequency domain after $z$-score normalization.

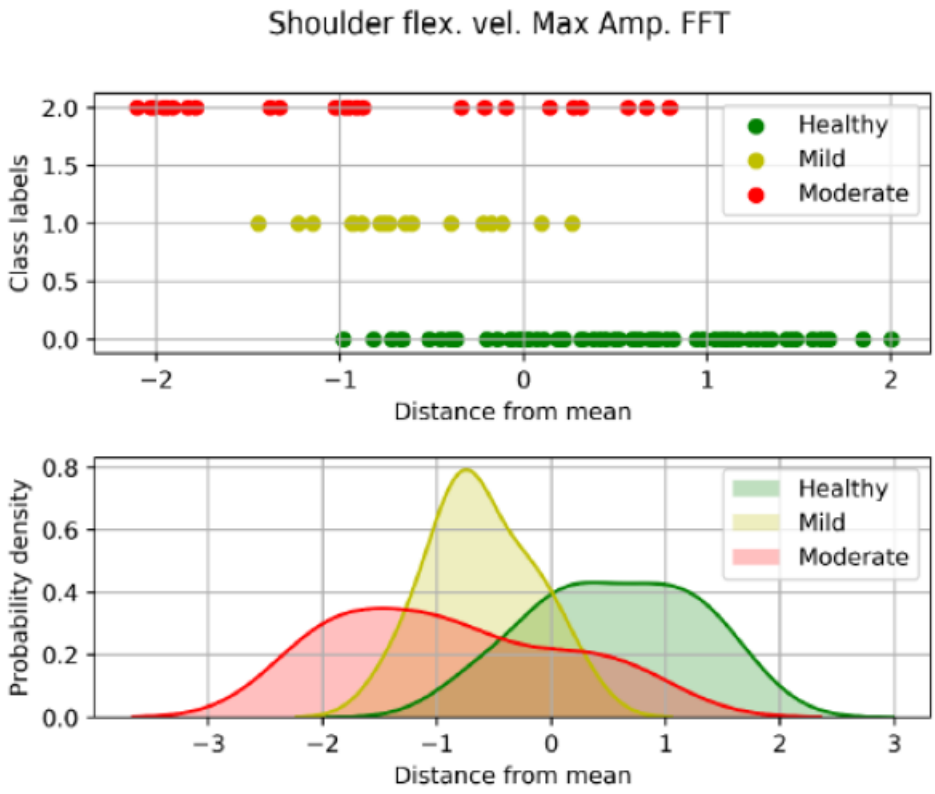



**Figure 40: Kernel desnity estimation used to estimate the probaility desnity function of Shoulder flexion velocity maximum amplitude in the frequency domain.**

The kernel density estimation helps to show how each feature separates the participant population. The feature shown in Figure 40 shows that this feature does separate participants based on stroke severity to an extent but it can't accurately classify the participants on its own. An ideal feature would have a probability density function for each healthy, mild, and moderate/severe participant population that does not overlap. A feature selection method that can find an optimal subset of features for classification is needed, that is, a subset of features that when combined can be used to accurately classify all participants.

### 5.5.2 Feature Selection Algorithms

A couple of problems arise when classifying stroke severity from the large volume of features used in this dissertation. The first one being that if it were desired to keep all 221 features, more data would be needed to avoid overfitting the model or making it too specific to the sample size that it was tested on (same situation shown in the previous section in Figure 36). This would mean that future participants that use the system would more likely be outliers to the SVM classifier if they slightly deviate from the training set trials.

Another problem is that more data is needed as the size of the feature set grows. Simon (Simon R., 2003) states that if the number of features significantly exceeds the number of observations (sample points), the classification problem becomes harder and overfitting will occur (machine learning model is not robust to noise in features). This is commonly known as the high dimensional, low sample size problem. Since data is limited for this study, the goal was to define a large set of features and then select an optimal subset of features that classify stroke. The workflow of how the features were narrowed down is depicted by the Venn diagram in Figure 41.

The total feature set included every candidate feature defined using the methods outlined in Aim 1. These features were chosen based on their potential to be used in a classification process of stroke severity, and an iterative process was used to search for a set of features that maximized the accuracy of the prediction process. An initial feature set was created and its stroke severity predictability tested in a rigorous machine learning feature selection process (to be explained in section 5.6). If the through testing of that feature set the F1 score was below 0.95, then the goal was not met, so new features had to be found. Rather than keep all features, the least valuable features were discarded. The features that were chosen as the top 20 stroke severity predictor for any given task were kept in the candidate feature set and the remaining features were discarded. Aim 1 was then revisited to define new features, and the feature selection process was repeated with least valuable features being discarded each time. Finally, that process resulted in a set of final features that are the most useful in the classification process.

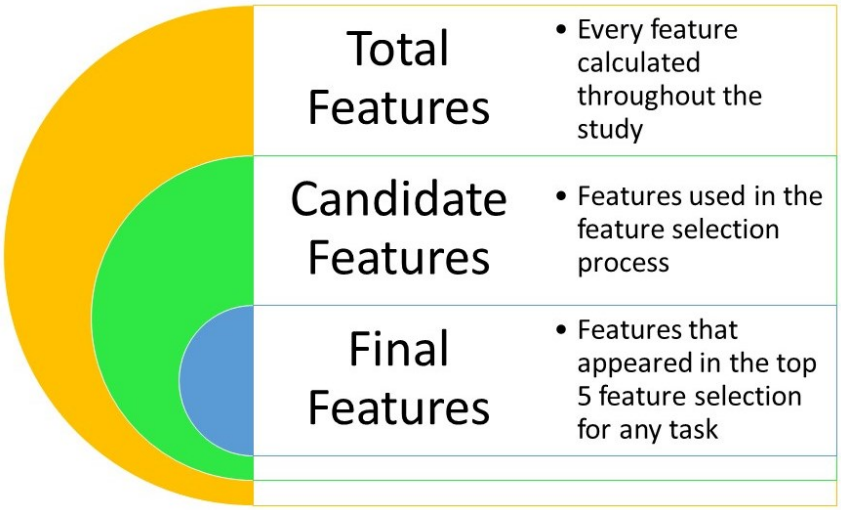



**Figure 41: The Venn diagram shows how the features were narrowed down to the top features that can classify stroke severity.**

Figure 41 illustrates the subsets of features. The total features are all features created for testing in the feature selection process. Candidate features are any feature that survived the process and were not discarded along the way. Final features will be explained in more detail in section 5.6 and are those that were finally chosen for stroke severity prediction.

The next section overviews several feature selection methods that can be used for machine learning problems.

#### 5.5.2.1 Filtering methods

A simple approach is to use filtering methods that observe the correlations between features. One filter method is to apply the Pearson correlation between each feature and the class labels. From there the features with the highest correlations with the class labels are kept. The Pearson correlation is not ideal in most real-life applications due to its assumption that the correlation between the two variables is linear. A nonlinear equivalent is the mutual information (MI) equation that extends the definition of entropy (in the information theory field) to find the correlation between two variables. The standard definition of entropy within the field is shown in equation (5-20).

$$S = -\sum_i P_i \log(\mathrm{P_i}) \tag{5-20}$$

where $P_i$ is the probability that the *i*th event occurs (Wang Q., 2019). The entropy shows that when an event has a low probability, it has more information. Equations (5-21) – (5-23) demonstrates how MI is calculated.

$$I(Y,X) = H(Y) - H(Y|X) \tag{5-21}$$

$$H(Y) = -\sum_{y \in Y} P(y) log P(y) \tag{5-22}$$

$$H(Y) = -\sum_{x \in X} \sum_{y \in Y} P(X|Y) log P(X|Y) \tag{5-23}$$

where $I(Y,X)$ is the MI between the class labels Y and features X. $H(Y)$ and $H(Y|X)$ are the entropy of the class labels and the conditional entropy of the class label's dependence on the features respectively. MI will be zero if Y and X are independent and greater than zero if they are dependent. Using this method to down-select features, a subset of features can be selected that maximizes the MI. Although these methods are computationally inexpensive, it is not guaranteed that the subset selected will obtain the optimal model accuracy.

#### 5.5.2.2 Embedded methods

The embedded feature selection methods are an optimization approaches to select the best features for the machine learning model. The common algorithms are the least absolute shrinkage and selection operator (LASSO) and Ridge regression (James G., 2013). The equations for LASSO and Ridge are included below.

$$\text{Lasso regression: } \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left( Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ji} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^2 \qquad \textbf{(5-24)}$$

$$\text{Ridge regression: } \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left( Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ji} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \qquad \textbf{(5-25)}$$

where $\beta$ is the coefficient for the linear regression and $\lambda$ is a weight for the additional constraint on the coefficients. Both methods add constraints to the linear regression objective function that keeps the coefficients from becoming large to avoid overfitting. Lasso accounts for the L1 norm between points while Ridge uses the L2 norm. The features that converge the coefficients closest to zero are selected for the machine learning model.

Another algorithm is the max-relevancy min-redundancy algorithm. This extends the MI method by also finding the MI between the currently selected features with the feature being observed at the optimization iteration. This algorithm selects a subset of features that have a maximum correlation with the class labels and a minimum correlation with other features selected. One drawback of the embedded method is that it cannot be guaranteed to find an optimal subset of features for the model since the optimization is not based directly on the model performance (such as maximizing the F1 score for the case of this study).

#### 5.5.2.3 Wrapper methods

Wrapper methods try to fill in the gap between the filtering and embedded methods by running an optimization that incorporates the chosen machine learning algorithm. The goal is to find a subset of features that trains the machine learning model to obtain optimal classification accuracy. The common algorithms include the sequential forward selection (SFS) and backward feature selection (SBS) methods. The forward selection method starts with an empty set of features and adds one feature at a time until k features (determined by the user) are obtained that optimizes the model accuracy. The process for sequential forward selection is below:

**Sequential Forward Selection Pseudo Code**

1.Create an empty set: $Y_k = \{\emptyset\},\ k = 0$

2.Select best remaining feature: $x^+ = \underset{x^+ \in Y_k}{\operatorname{argmax}}[J(Y_k + x^+)]$

3.If $J\big((Y_k + x^+)\big) > J\big((Y_k)\big)$

a.Update $Y_{K+1} = Y_k + x^+$

b.$k = k + 1$

c.Go back to step 2

In the algorithm presented above, $J(Y_k + x^+)$ is the accuracy of the machine learning model when feature $x^+$ is added to the feature array. This process is repeated until the optimal $k$ (selected by the user) features are found.

The backward sequential algorithm works in reverse, starting with a full set of features and obtaining the optimal accuracy by removing features from the set. The problem with SFS is that features added to the subset cannot be removed which creates a suboptimal selection (Smith A., 2014). One workaround is the sequential forward floating selection (SFFS) where an additional exclusion step is included. In this exclusion step, a feature is removed only if it increases the machine learning model performance. A flow chart of the SFFS is included in Figure 42.

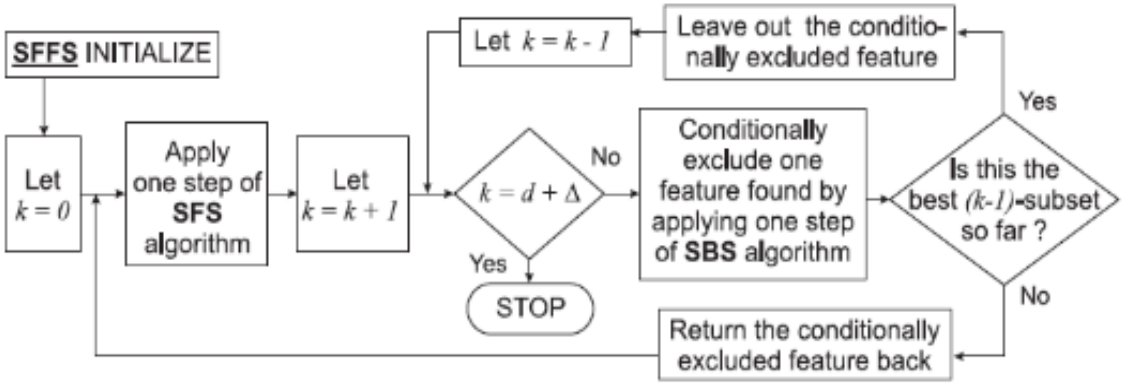



**Figure 42: A flowchart showing the Sequential Floating Forward Selection Process (Somol P., 2014).**

For this study, the SFFS was used as the feature selection method. To remove any additional redundancies, the Pearson correlation equation was used to drop one from a pair of features that had a Pearson correlation coefficient above 0.75. This allowed other features to be explored since high-performance features can be highly correlated with many other features.

## 5.6 Application of the Feature Selection Method

---

*Measure of success: Classification of stroke severity using motion features [Desired outcome: F1 score 0.95 – 1].*

---

Using the SFFS for feature selection and one vs one SVM as a machine learning algorithm, stroke severity was classified for all the reaching tasks discussed in chapter 3.0. The feature selection process was performed for each iteration of Candidate Feature sets that were created in Aim 1. As explained in Section 5.5.2, the Candidate Feature set was redefined (more features were created and added) when the F1 score did not reach at least 0.95 and features were not selected as a top 20 feature at least once throughout all the tasks. The results shown in this section narrow down the final set of Candidate Features (154 features, depicted in Figure 43 and listed in Appendix B) to the top five features for any given task (a top five feature is any feature that is one of the five most selected features for a given task during the feature selection process).

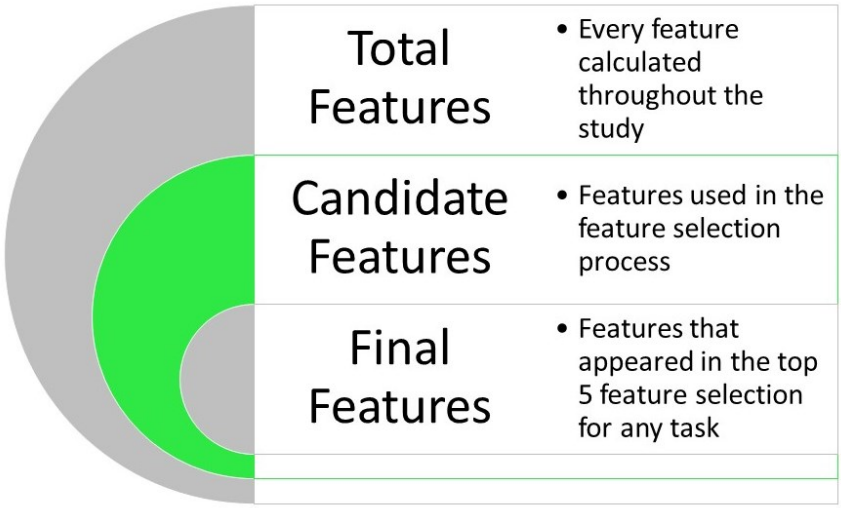



**Figure 43: The Feature Venn diagram highlighting the candidate feature as the current step to the feature selection process.**

One change from the initial procedure described above was the stroke participant distribution. All 10 healthy participants were recruited but there was some difficulty in recruiting severe participants to the study. In many cases, severe participants recruited from the Pitt+Me database had lost all mobility in their affected arm, so they were unable to complete the tasks. Also, mild participants with an FMUE score over 60 were almost indistinguishable from healthy participants and caused them to be misclassified, so they were moved to the healthy class for the purposes of this study. The patient distribution is shown in Figure 44.

| # of participants | Stroke severity |
|---|---|
| 8 | mild |
| 8 | moderate |
| 1 | Severe |

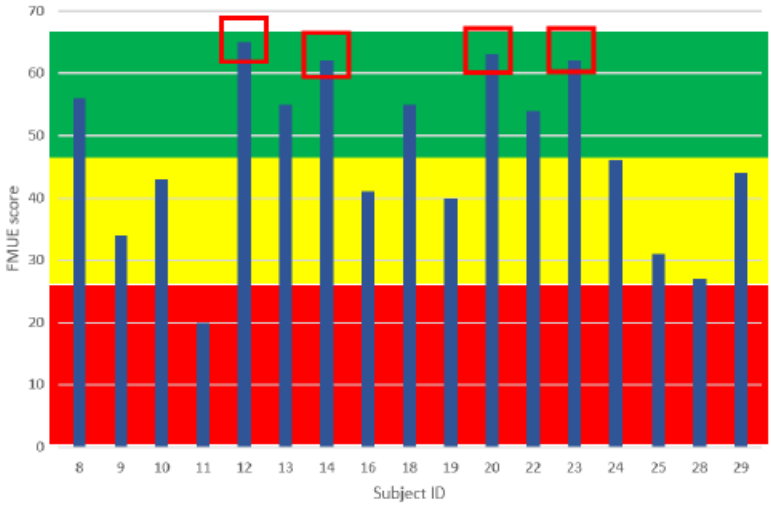



**Figure 44: Stroke participants' distribution in the study based on their FMUE scores. The mild participants highlighted with a red box were moved to the healthy class.**

The lack of severe stroke subjects and the reclassification of highly performing mild stroke severity patients created a further imbalance in the dataset so the SMOTE interpolation was heavily applied to the mild class since it was reduced to four participants. It is also important to note, due to only having one severe participant who completed the study, it was decided to place that subject into the moderate population for classification.

Through the data analysis process, the results showed that including all five trials for each subject provided the model with redundant information. Another way to think about this is that everyone has their signature motion pattern which will rarely deviate in such a short amount of time. Therefore, additional trials add no new information. To test this theory, the Pearson

correlation was calculated between the feature set of each trial. The results can be seen in the heat map in Figure 45.

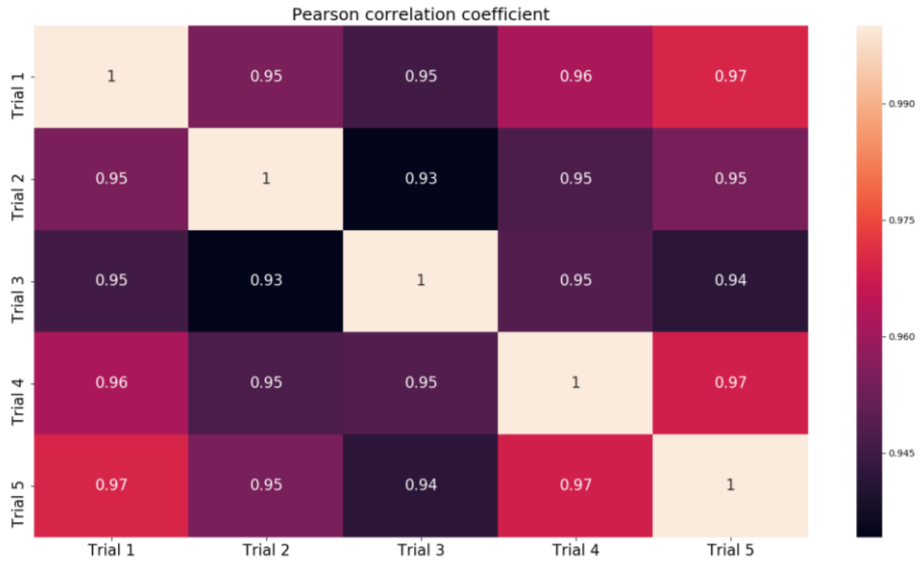



**Figure 45: Pearson correlation between feature vectors for each trial of a single participant. The results show that the features extracted from each trial are nearly identical.**

Figure 45 shows that all 5 trials presented essentially the same information in the feature space since the Pearson correlation was greater than 0.95. This was a problem when the data was split into training and test sets since the same point to train the data cannot be used to test the accuracy of the machine learning model. To combat this problem, only one trial from each subject was randomly selected to represent the patient population.

The process of determining the subset of features that are most useful for stroke severity classification, the Final Features as shown in Figure 46, is described in this section.

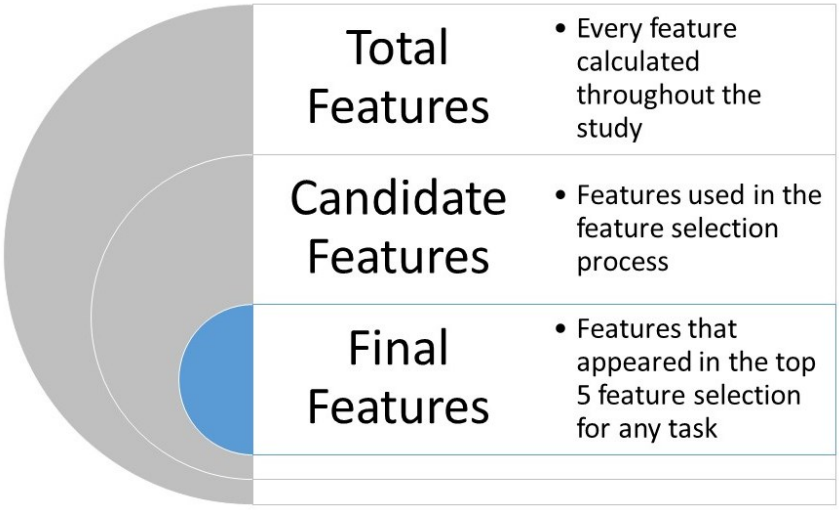



**Figure 46: The Venn diagram highlights the final subset of features used in Aim 3.**

The current population was randomly split into training data (70% of the population) and test data (30% of the population). For the feature selection process, the random selection was done for each of the 150 iterations described earlier in this chapter. The SFFS was used to select the top 5 features in the model based on the average F1 score for all 3 classes.

For the short pointing target 5 task, Figure 47 – Figure 50 show example results of the number of times each feature was chosen among the final 5 features for an iteration of the SFFS process, for each of the four kernels. Note that for clarity, only the 20 most-selected features are

shown in the graphs, with the top 5 most-selected highlighted in red. For any given task, more features than the 20 shown in the charts were chosen at least once during the 150 iterations, but only the top 20 are shown to simplify the charts.

For the remainder of this section, the results for only the short pointing target 5 task are shown, as examples. Appendix B contains the full results for all tasks.

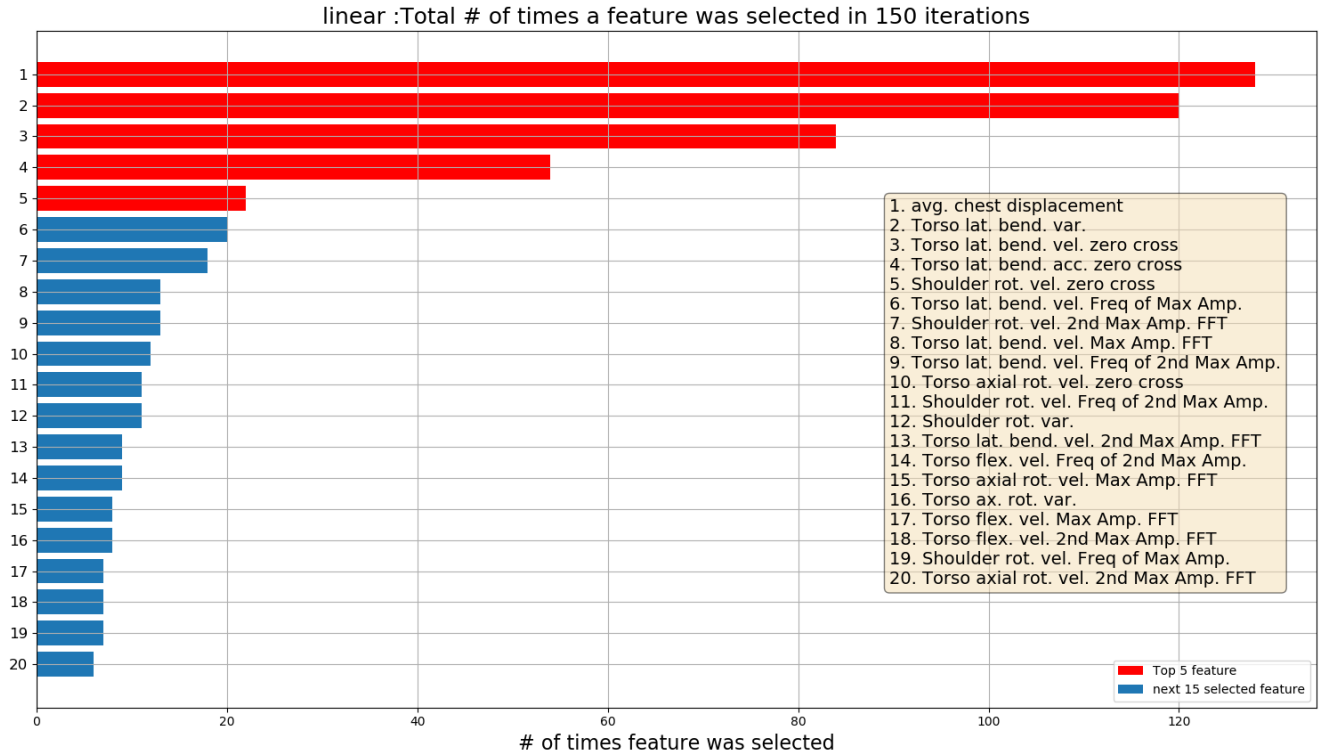



**Figure 47: The top 20 features (out of 154 features) for classifying short pointing target 5 task, selected by the linear kernel .**

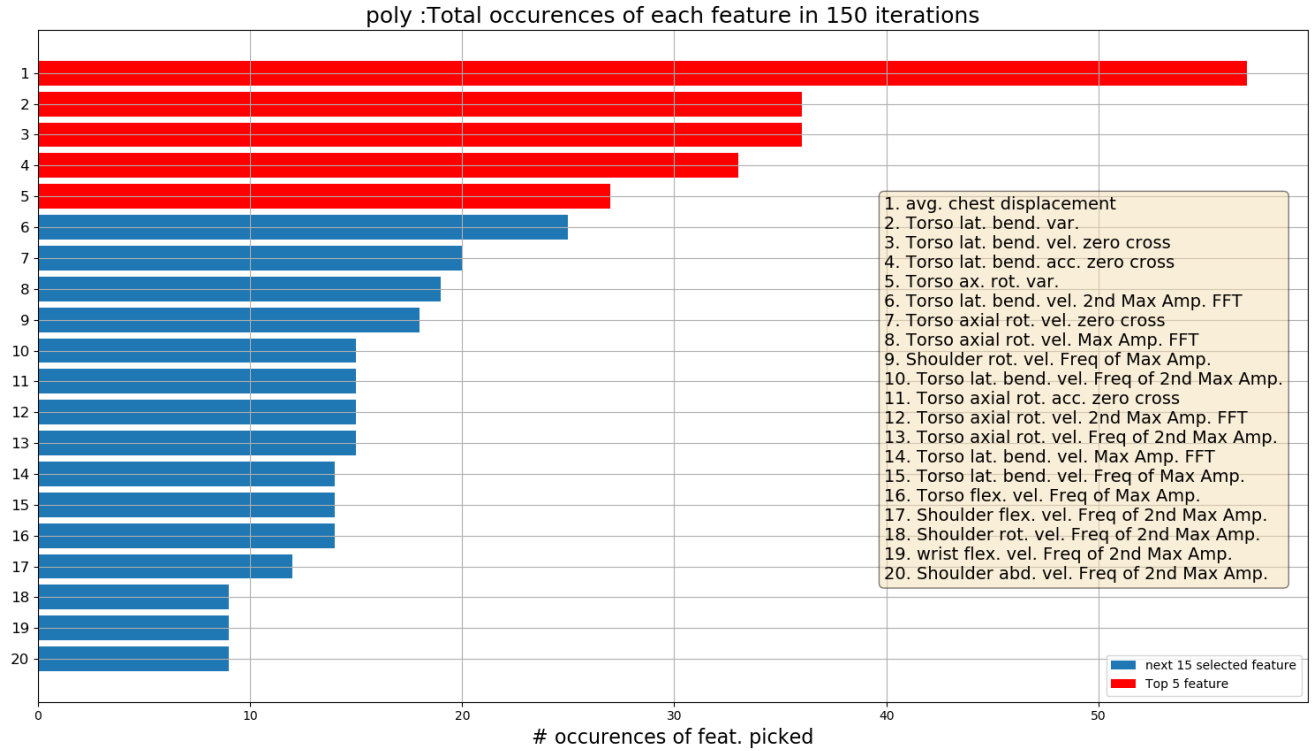



**Figure 48: The top 20 features (out of 154 Candidate Features) for classifying short pointing target 5 task, selected by the polynomial kernel.**

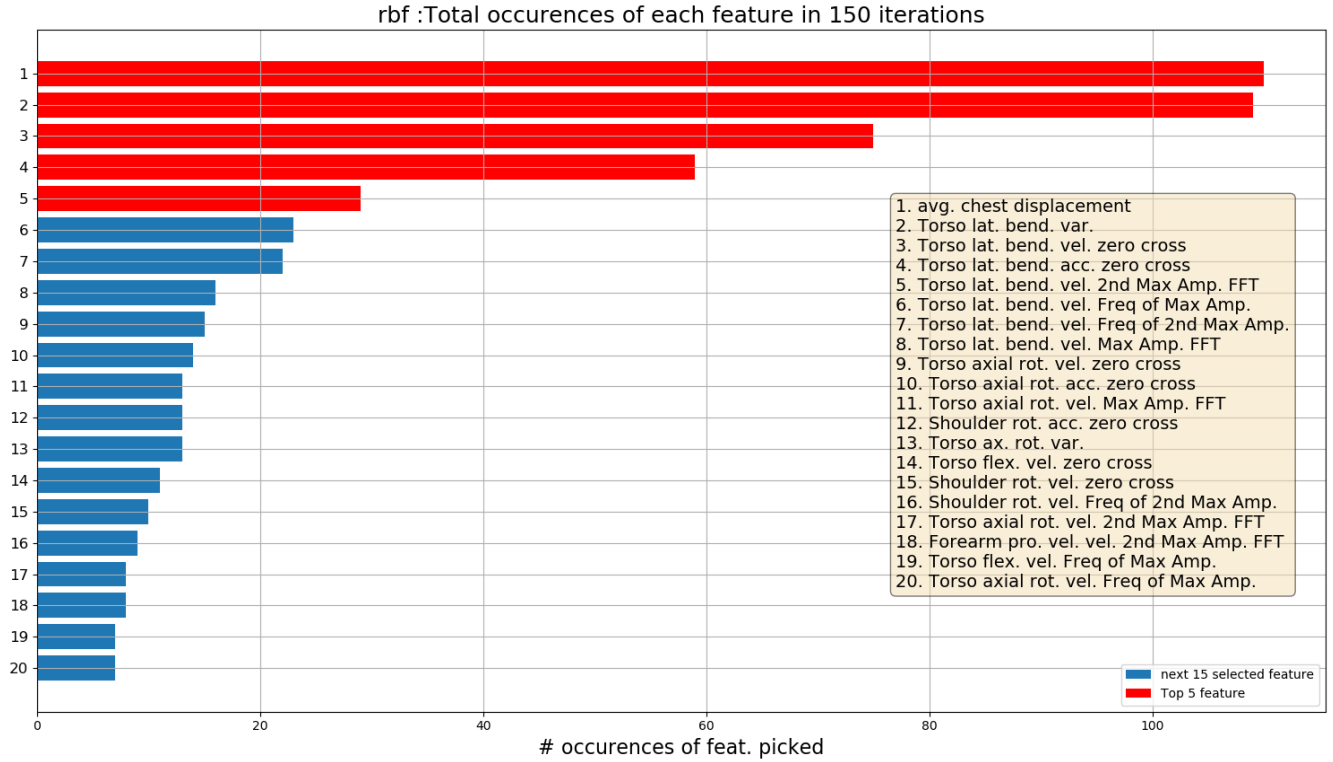



**Figure 49: The top 20 features (out of 154 features) for classifying short pointing target 5 task, selected by the RBF kernel .**

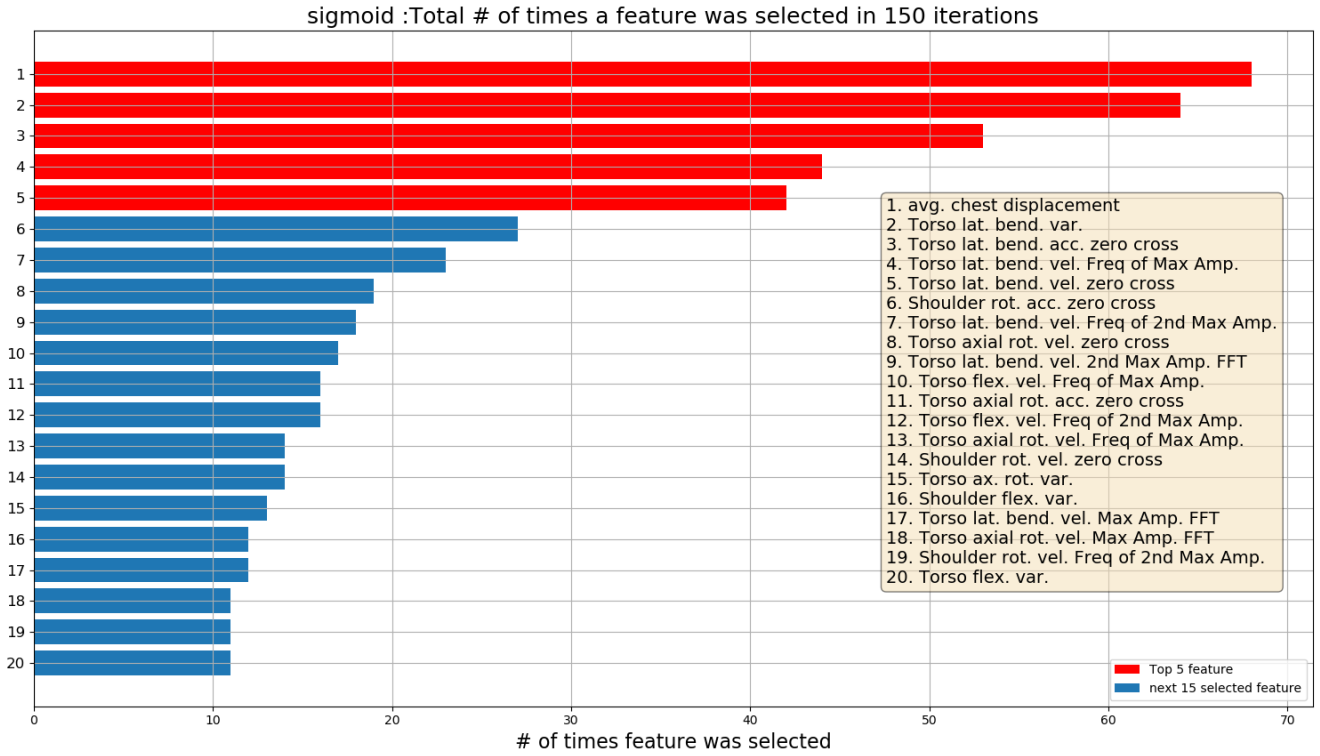



**Figure 50: The top 20 features (out of 154 features) for classifying short pointing target 5 task, selected by the sigmoid kernel .**

As described above, the SFFS method relies on the F1 score when any given set of 5 features selected with the training data are then evaluated for performance (F1 score) using the test data. A snapshot of the F1 score over 150 iterations can be seen for the short pointing target 5 task in Figure 51.

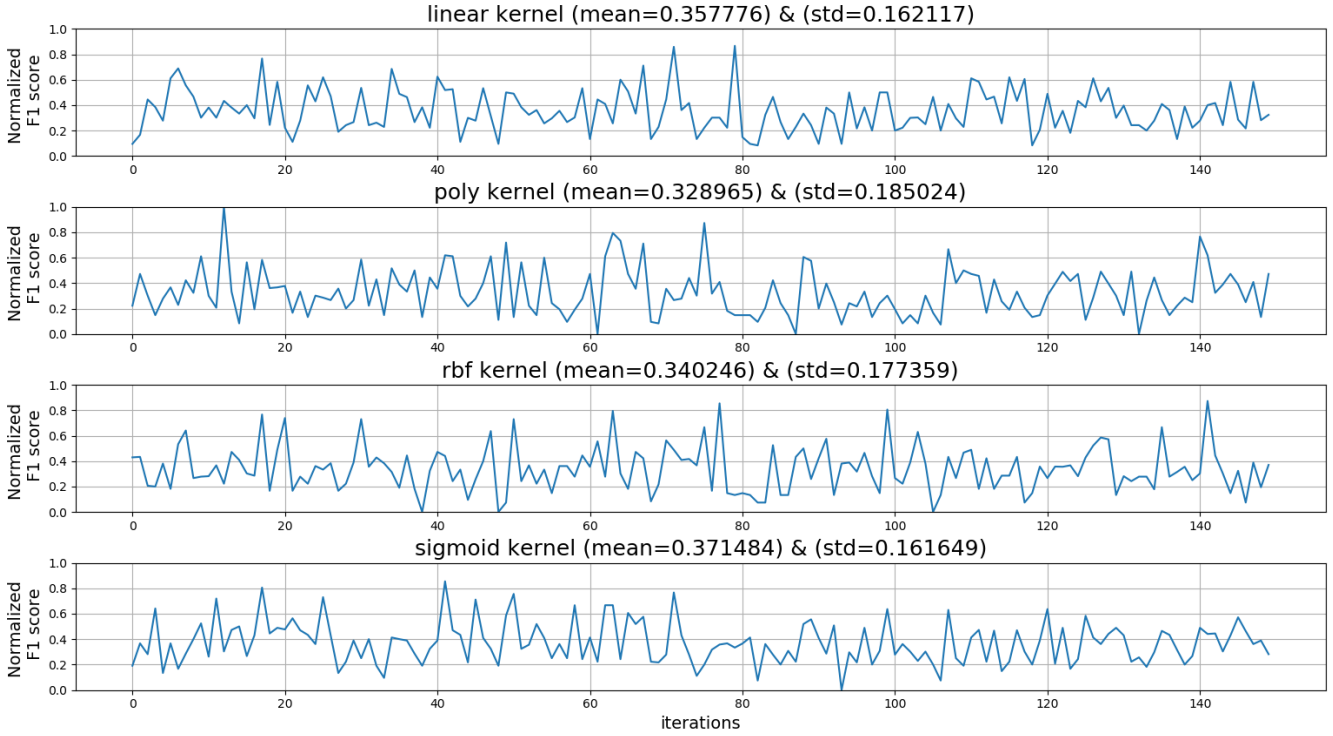



**Figure 51: F1 score over 150 iterations of randomly selecting trials and splitting data (train/test) for short pointing target 5 task. The mean and standard deviation of the F1 score is shown in the title of the plots.**

Because the fluctuation in the F1 score shown in Figure 51 is highly dependent on the 70% training data selected to train the model, this can result in sensitivity of the results. As mentioned in section 5.3, SMOTE was used to upsample the minority classes (either mild or moderate/severe participants who could not finish the task). Due to the limited amount of points to interpolate within some tasks, the minority class could not be upsampled to exactly match the healthy class. The data limitation also limits the area in the feature space that the SVM algorithm can explore when training since the interpolated points lie between the few actual points defined.

For iterations where the F1 score dipped below 0.6, it was caused by a larger stroke population in the test set than the training set and vice versa when the performance was high. The confusion matrices in Figure 52 – Figure 55 reinforces that the algorithm has a flaw in classifying the stroke participants. In the perfect scenario, the confusion matrix would be an identity matrix showing no misclassifications.

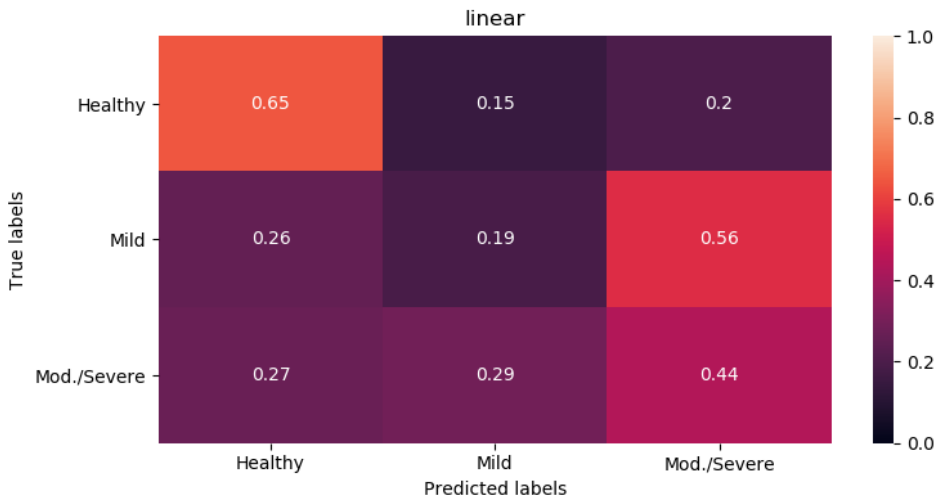



**Figure 52: Confusion matrix showing the predicted values vs the true values for the linear kernel.**

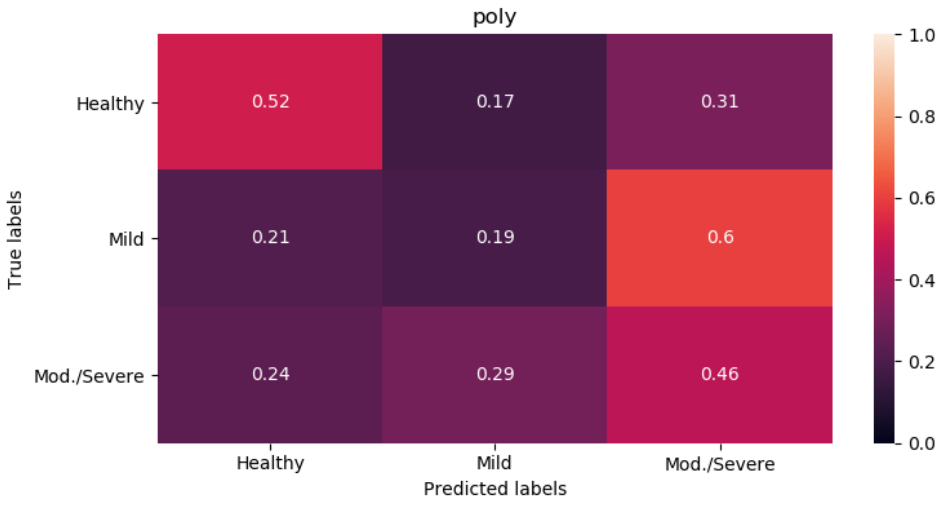



**Figure 53: Confusion matrix showing the predicted values vs the true values for the polynomial kernel.**

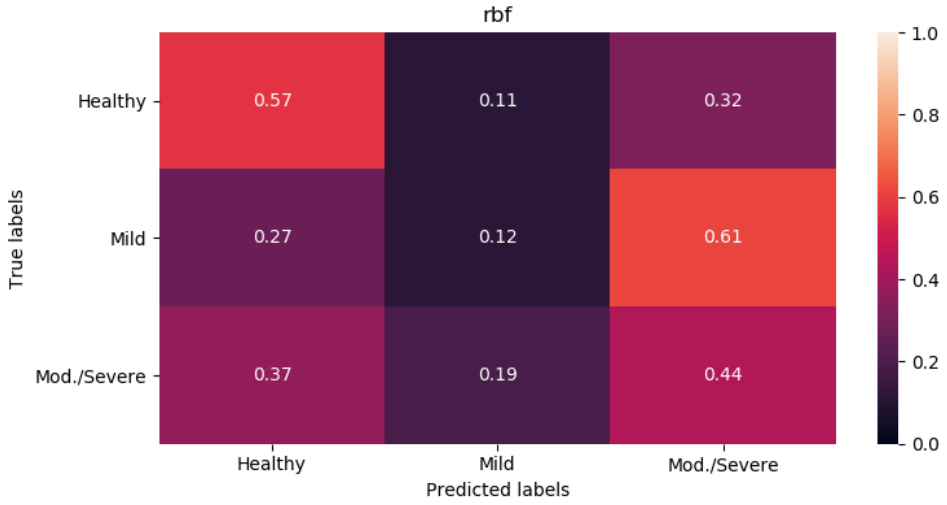



**Figure 54: Confusion matrix showing the predicted values vs the true values for the rbf kernel.**

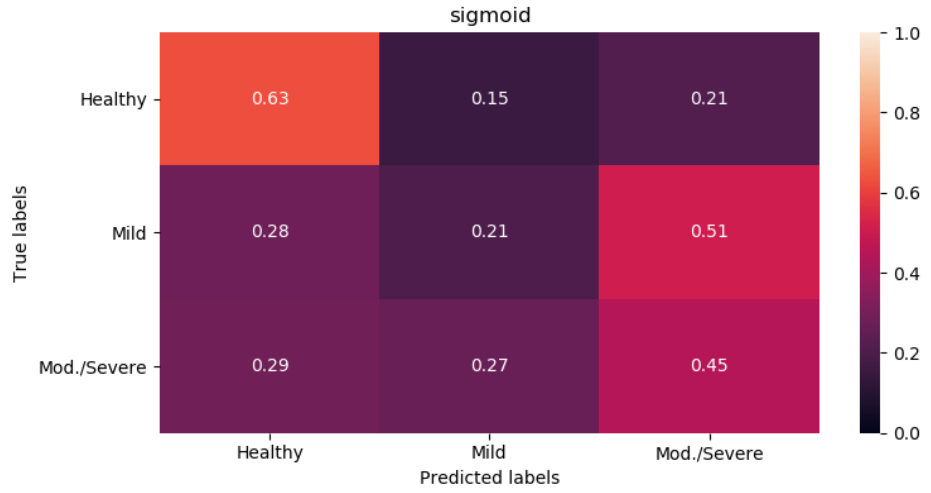



**Figure 55: Confusion matrix showing the predicted values vs the true values for the sigmoid kernel.**

Based on Figure 52 – Figure 55, the healthy participants are classified correctly at least 65% of the time since they contain most of the sample points in the training data. While the stroke participants' classification is almost evenly distributed in all three classes. The moderate/severe class had a higher classification accuracy than the mild class since it had a large participant population.

To see if the F1 score could be improved, the grid search method was used to tune each SVM kernel parameter to adjust the decision boundaries. The top 5 features defined earlier were used during the tuning process. The performance of the tuned parameters was tested in the 150-iteration process again. Figure 56 – Figure 59 shows the results for the tuned parameters with each point representing the average F1 score and the error bars representing the standard deviation from the average. The ideal tuning result would be an F1 score close to 1 (e.g. >0.9) with a narrow error

bar (e.g. <0.05). This would ensure that the machine learning model was accurate and stable (able to predict stroke severity accurately given any 70% patient population as training data).

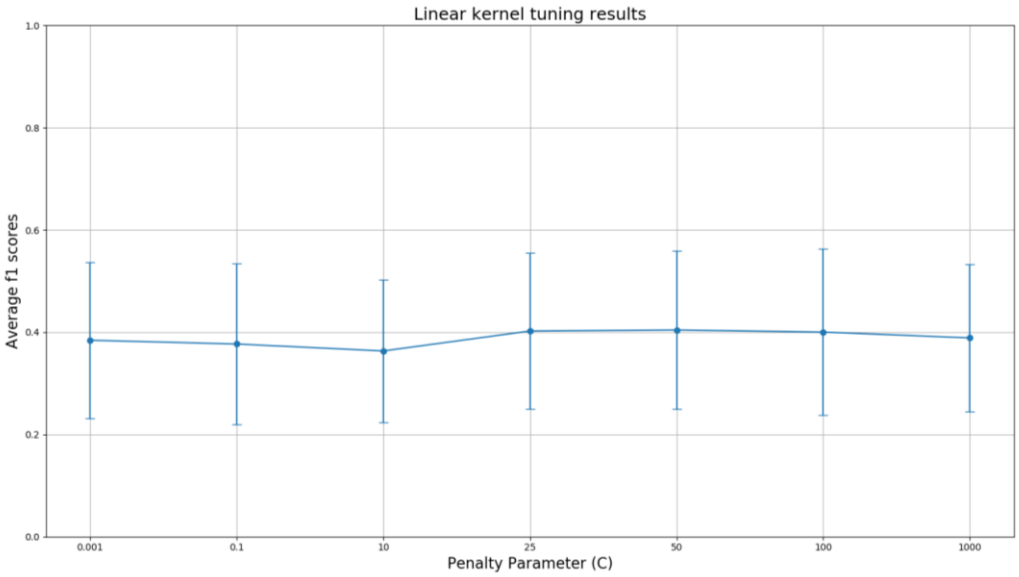



**Figure 56: Average F1 score over 150 iterations for each C parameter value applied to the linear kernel. The error bars represent one standard deviation from the average. Gamma does not apply for the linear kernel.**

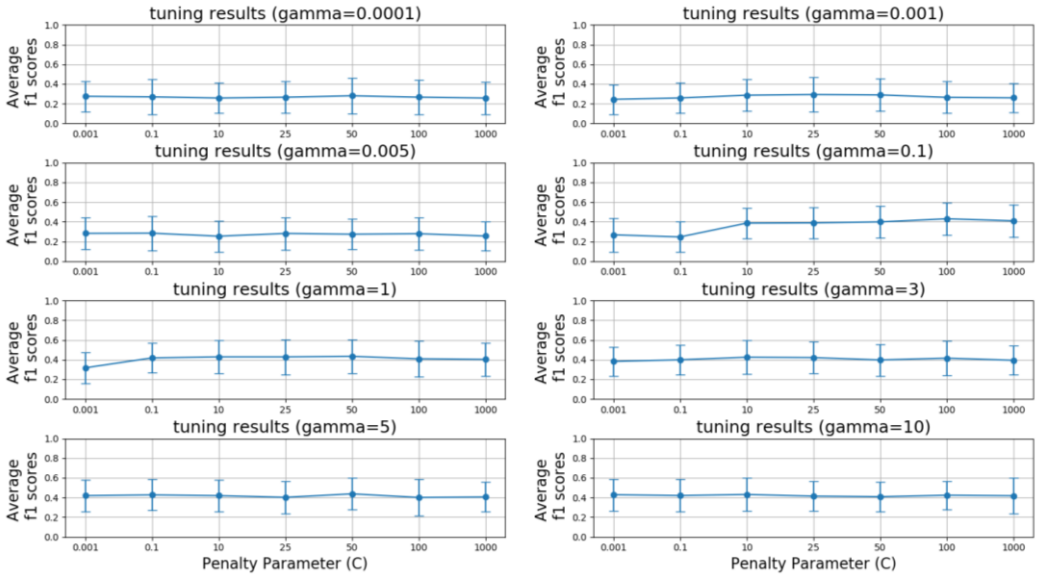



**Figure 57: Average F1 score over 150 iterations for each C parameter value applied to the polynomial kernel. Each subplot represents a fixed gamma with a varying C. The error bars represent one standard deviation from the average.**

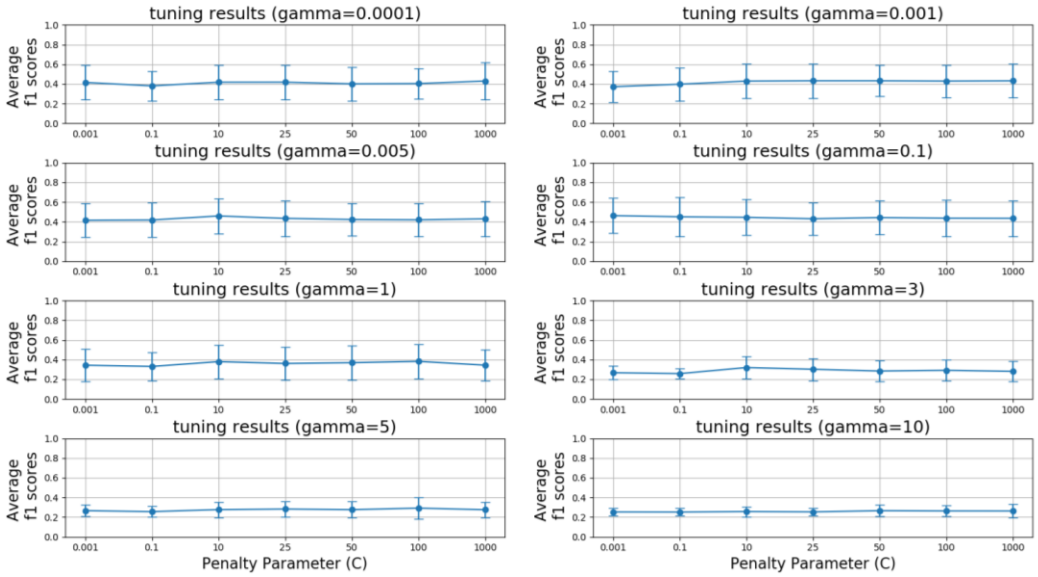



**Figure 58: Average F1 score over 150 iterations for each C parameter value applied to the RBF kernel. Each subplot represents a fixed gamma with a varying C. The error bars represent one standard deviation from the average.**

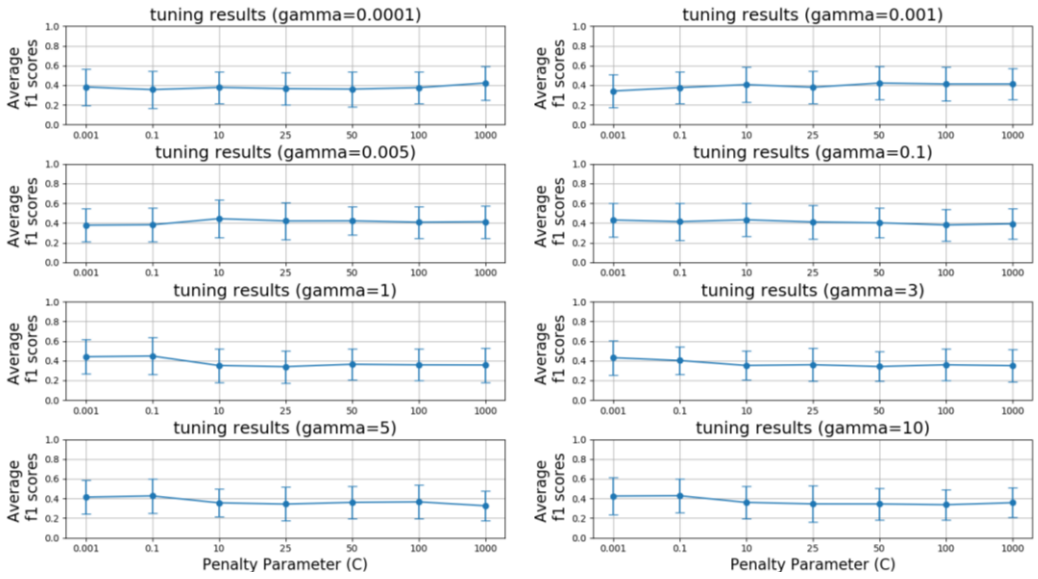



**Figure 59: Average F1 score over 150 iterations for each C parameter value applied to the sigmoid kernel. Each subplot represents a fixed gamma with a varying C. The error bars represent one standard deviation from the average.**

Tuning the model parameters slightly improved the performance of each kernel when the best parameters were chosen but there was still room for improvement since the F1 score was not close to 1 (for example, the best F1 score with tuning was 0.462 for $C$=0.001 and gamma=0.0001 in the RBF kernel). The obvious solution was to increase the number of features chosen during the feature selection but due to the limited size in the dataset, this was not an option since it would introduce overfitting as stated earlier.

Another approach to improve the process was to use a combine the predictions between all the kernels. Instead of comparing the kernels with one another, they can be merged using a voting

system. This is known as a machine learning ensemble method in which multiple algorithms are combined to build a single machine learning model with a better performance. Many ensemble methods exist such as stacking, boosting, bagging, and voting. The stacking and bagging ensemble methods could not be used since these algorithms split the data into smaller subsets of training and test groups. This requires a larger dataset than was available in this dissertation. Boosting was not used since it can introduce sensitivity to noise and outliers (overfitting), which eliminates the advantages of the SVM that was stated in section 2.1 (Mayr A., 2014). While the voting method uses a majority or weighted vote between multiple machine learning algorithms to make a final prediction. The selected ensemble method used in this dissertation is the voting ensemble algorithm and is illustrated in Figure 60.

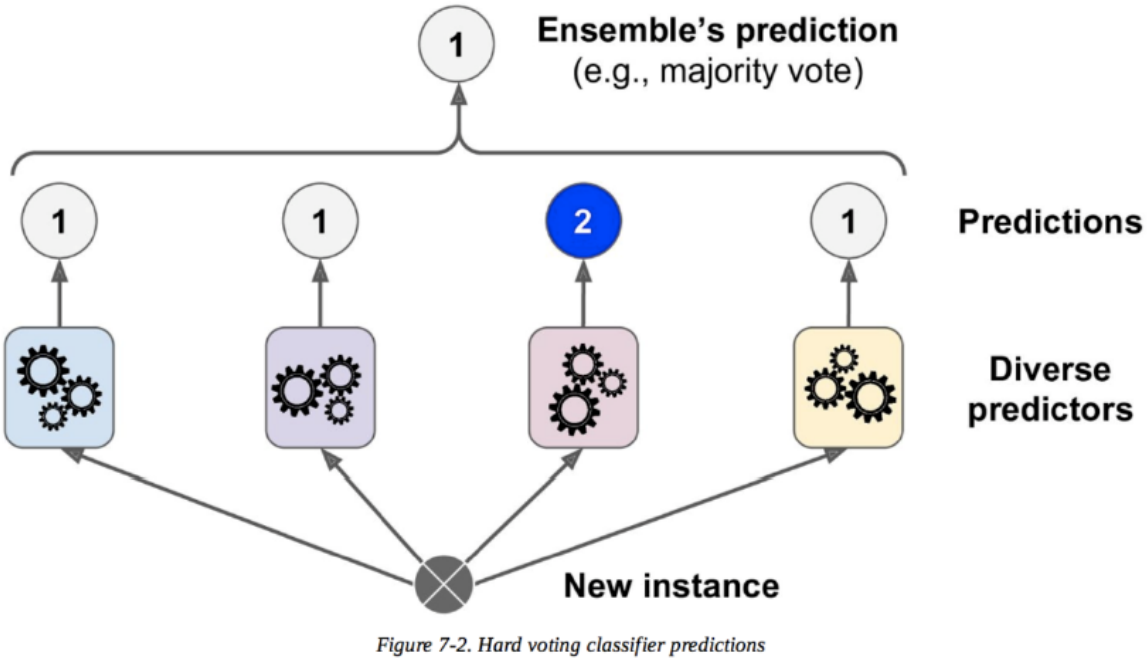



*Figure 7-2. Hard voting classifier predictions*

**Figure 60: Illustration of the ensemble classifier voting method. Each classifier votes the prediction and the majority vote is the final classification (Géron, 2017).**

As shown in Figure 60, all the classifiers (in this case all the linear, polynomial, RBF and sigmoid kernels used in SVM) make a prediction and the majority vote becomes the final classification. The results of this ensemble method where all four kernels made a vote is shown in Figure 61 and Figure 62.

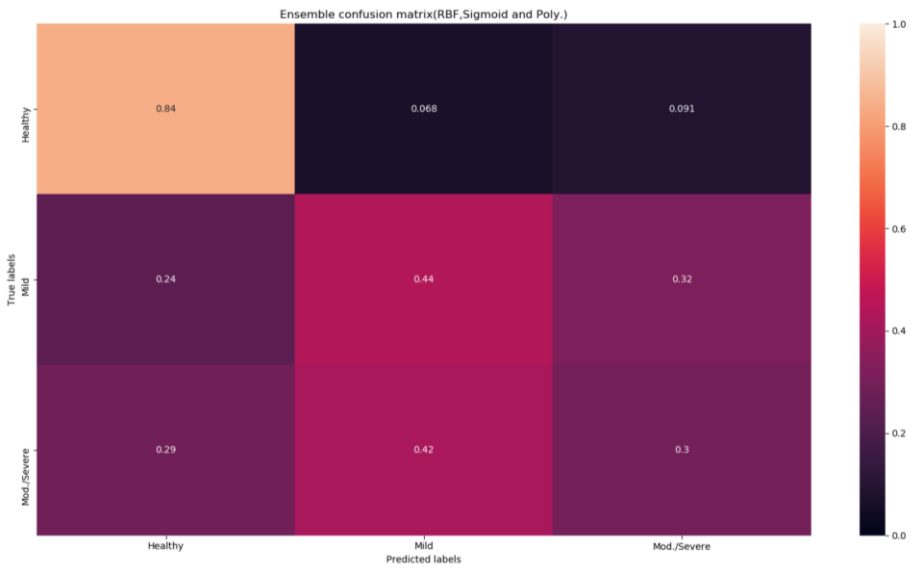



**Figure 61: Ensemble confusion matrix for short pointing target 5.**

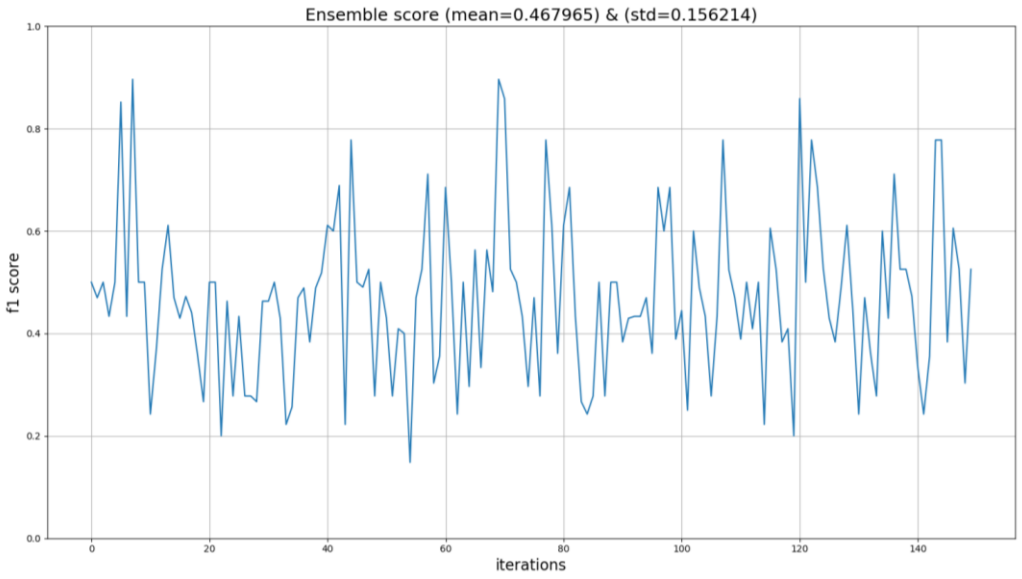



**Figure 62: Ensemble classification for short pointing target 5 during the 150 iteration process. The mean and standard deviation of the F1 score is shown in the title of the plot.**

Figure 61 shows that by using the ensemble method, the classification accuracy has increased by 12% and 5% for the healthy and mild classes respectively while the classification accuracy decreased 12% for the moderate/severe class compared to the best results shown in Figure 52 – Figure 55. Looking at the results of the F1 score throughout the 150 iterations in Figure 62, the F1 score still fluctuates but the overall average F1 score has increased by about 10% compared to the individual kernels before tuning and increased about 0.6% for the RBF tuned kernel.

The results shown in this section have focused on the short pointing task 5. Other intuitively "easy" tasks, such as short tasks (40% of arm length) showed similar general classification results.

It is interesting, however, that for harder tasks such as when the target system was set to be 80% of the participant's arm length, the classification improved. For example, Figure 63 shows the confusion chart for the ensemble classifier for the far grasp target 3 task.

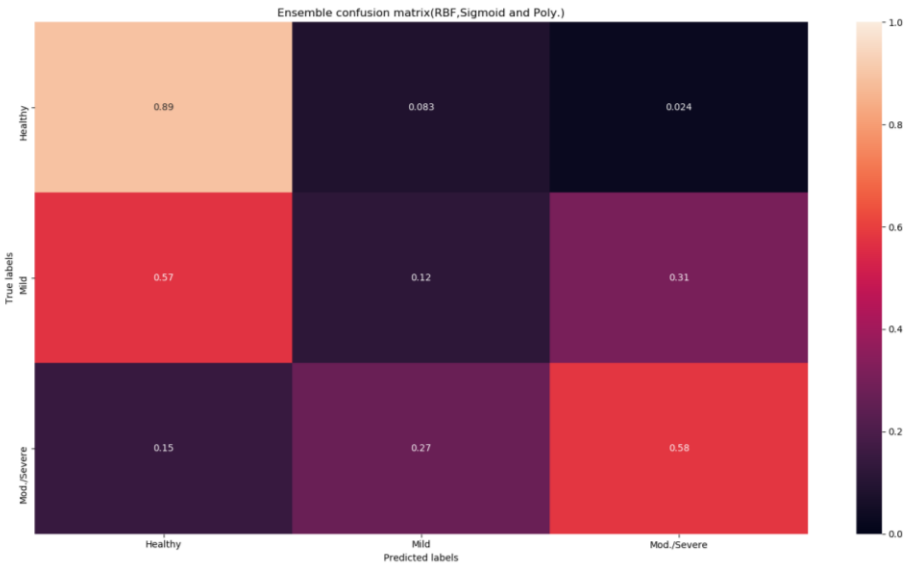



**Figure 63: Confusion chart for the ensemble classifier used for far grasp target 3 task.**

The results in Figure 63 show that the healthy class was easier to identify with the ensemble classifier than the mild and moderate/severe classes. The increase in accuracy was caused by the increase in the difficulty of the task which shows the harder the task the easier it was to detect impairment. It is important to note, that the stroke sample size for severe patients dropped for this task (2 severe patients dropped due to incompletion). Since the severe patients' features deviate

further from healthy due to their difficult to complete the task, the classification accuracy still improved. This did affect the performance for mild patients due to the nature of the task which highlights high range of motion for some patients and not others. The SMOTE interpolation could have also contributed to the noise of mild stroke patient classification by creating child points between mild patients that are close to full recovery and patients still recovering. Some tasks could not be classified due to the severe participants not being able to complete it.

The work done for Aim 2 established a machine learning framework that predicts a patient's stroke severity based on their upper extremity kinematics and kinetics. This system will be beneficial for stroke patients looking to receive an immediate evaluation of their stroke severity and rehabilitation progression without waiting to receive feedback from a clinical expert. A list of the selected features for each task is shown in Appendix B. The remainder of this section discusses how the measure of success was met.

## 5.7 Aim 2: Measure of Success Achievements

With the completion of Aim 2, the overall measure of success was not met for each task where the F1 score had to fall between 0.95 – 1. The main reason for this shortcoming was due to the lack of data collected which made it difficult to predict stroke severity for the minority class (usually mild stroke patients for easy tasks and moderate/severe stroke patients for harder tasks). Although the SMOTE method was used to emulate a larger patient population, there were cases where only two mild patients completed the tasks so there was not much variability in the interpolated patients. In some of the hard tasks, for example, far rotation for target 8, not a single severe participant could complete the task.

This lack of data limited the predictability of the stroke patients' severity and a more balanced population of healthy and classes of stroke patients needs to be represented in future studies to complete this system. At a minimum, 10 healthy, 10 mild and 10 moderate/severe patients need to be recorded for each reaching tasks, but a much larger data set would greatly improve the results. This will ensure that 70% of the training data will adequately represent the general patient population for the SVM to predict stroke severity accurately for the remaining 30% of the test data. Additional subjects should be collected if the model continues to be unstable (F1 score is not >0.95 for all iterations of the training data as in Figure 62).

Although the measure of success was not met, the overall framework to create a machine-learning algorithm to predict stroke severity is still an important contribution to an at-home rehabilitation system that the next researcher can deploy with a larger dataset. Given the increasing separation between healthy and stroke patients with harder tasks, the methods developed in this aim were considered to be acceptable for use in Aim 3 to establish a method of goal-oriented rehabilitation through a reaching task hierarchy. The top 5 features from each kernel within each task, were used to define the task hierarchy. These were the final subset of features selected in this dissertation as shown in Appendix B.

# 6.0 Aim 3: Define Individualized Task Difficulty Hierarchy

This chapter provides an overview of how the series of tasks described in chapter 3.0 can be ranked based on difficulty. The goal is to use this system to provide participants with a personalized rehabilitation program that mimics the advice given by a trained clinician. To see how the task hierarchy changes between stroke participants, results are shown for one stroke participant in each class (mild, moderate and severe).

Section 6.1 details how motor learning theory can be used to regain mobility in the affected arm and the broader impact of telerehabilitation for the stroke community. Section 6.2 describes how Euclidean distance in the feature space can be used to create a custom rehabilitation regime and measure progression. The summary of findings for the automated personalized task hierarchy that was created with the final features for each task is shown in section 6.3. Lastly, section 6.4 discusses how the measure of success was met for Aim 3.

## 6.1 Theory and Methods

From a patient's perspective, the application of the SVM algorithm to classify stroke severity is a useful tool if he or she would like to know their current state of impairment, but the next logical step is to facilitate motor recovery to regain the mobility that they once had. Langhorne (Langhorne P., 2011) details that the typical cycle that a patient experiences post-stroke involves (1) assessment, to identify and quantify the patient's needs (2) goal setting, to define realistic and attainable goals for improvement and (3) intervention, to assist in the achievement of goals.

The assessment stage is covered by the FMUE scale that monitors their current weakness and dyscoordination (the outline for automation of this process has been covered in Aims 1 and 2). Once they have been given an FMUE score their rehabilitation is based on their current stroke severity.

The current gold standard motor learning method for stroke rehabilitation is a repetitive task-oriented regimen where the clinician needs to define achievable tasks (2. Goal Setting) and correct impaired motion patterns (3. Intervention) (Winstein C.J., 2014). To properly execute the goal-setting and intervention steps, the clinician has to follow three objectives: (1) the task must be challenging enough to require new learning; (2) the training must progress in difficulty and the task must be iteratively adapted to real-world relevance; and (3) the task must not lead to rote repetition but repeated attempts to solve a problem. Clinicians usually focus on tasks that mimic common motions used throughout the day otherwise called activities of daily living (ADLs) to meet these three objectives.

Thant (Thant A.A., 2019) evaluated the effectiveness of task-oriented rehabilitation using a trained physical therapist to provide instructions. The activities of daily living assigned involved lifting a glass of water, moving crystal balls from a table to box, wiping down a table with a towel and combing their hair. The physical therapist would adjust the difficulty of each task through either changing the speed, distance and/or resistance based on the patient's ability. After two weeks of training (10 hours) and four weeks of training (20 hours), the average FMUE score increased 5.36 and 6.45 points from their base score, respectively.

Unfortunately, participants often do not follow this strict regime and become frustrated or lose motivation when applying the rehabilitation regime without the supervision of a clinician. A

system that provides the same expertise as a trained clinician is needed to keep the rehabilitation momentum consistent.

Telerehabilitation has been on the rise with the frequent innovations of computers and microelectromechanical systems (MEMS) sensors. Tsekleves (Tsekleves E., 2011) used the MEMS sensors in a Wii (Nintendo Co., Ltd) remote and Blender (open-source 3D environment software) for stroke rehabilitation at home. They were able to create a system that mimics a patient's full-body movement in a virtual environment. Their system is in the prototype stage, but the end goal is to relay the 3D avatar to the clinician to provide instructions remotely.

Buonocunto (Buonocunto P., 2017) takes it a step further by providing a MEMS sensor-based telerehabilitation system that introduces the patient with a series of tasks recommended by a physical therapist during an in-person visit. The patient can then perform a task with feedback from a 3D avatar. Afterward, a score is provided based on the amount the patient deviated from a fixed joint trajectory.

Although both systems show promise, they cannot rank tasks based on the patient's performance without the clinician remotely monitoring them. A data-driven method is needed that can automatically modify a series of tasks based on the patient's current ability.

This chapter outlines the process to create customizable instructions for each stroke patient. The goal is to rank the task described in chapter 3.0, using the data collected.

## 6.2 Establishing Task Hierarchy Using Euclidean Distance Between the Stroke and Healthy Feature Clusters

The purpose of creating the task hierarchy is to enable a stroke patient to start rehabilitation with a task that is challenging but relatively easy for them. It is anticipated that after mastering their first task, a patient will approach another, slightly more difficult task with more confidence. This will be an iterative process as a patient works their way through the series of tasks listed in chapter 3.0.

Since the rehabilitative goal is to have patients regain the motion patterns of healthy subjects, the difficulty of any given task for a particular subject was determined as a measure of how far a stroke subject's motions deviated from the motion patterns defined by the healthy subject population. The first step of this process was to define healthy motion patterns for each task in the feature space. The feature space for the given task was defined from the top features selected using the ensemble SVM classifier described in chapter 5.0. Using data from all the healthy participants the centroid for the healthy cluster is defined with the average of the feature set as shown below.

$$C_i^{Healthy} = \frac{1}{14}\sum_{j}^{14} z_{i,j} \tag{6-1}$$

$$C^{Healthy} = \begin{bmatrix} C_1^{Healthy} & \cdots & C_M^{Healthy} \end{bmatrix} \tag{6-2}$$

where $C_i^{Healthy}$ represents centroid for healthy for the ith feature that is an average across all 10 healthy participants and 4 mild participants with FMUE score over 60 (total of 14 participants),and $z_{i,j}$ represents the *i*th feature of participant *j* after it has been normalized using the *z*-score. The

final centroid vector in equation (6-2) has 1x$M$ dimensions where $M$ is the number of unique features chosen by the ensemble SVM classifier. As mentioned in section 5.6, $M$ changes based on the amount of overlap of features selected between kernels feature space can be 5 (when all four kernels pick the same 5 features) to 20 (when all four kernels pick unique features). Once the centroid is defined for each task, the hierarchy can be set up by calculating the Euclidean distance between a stroke participant and the healthy cluster centroid. An illustration of this hierarchy can be seen in Figure 64.

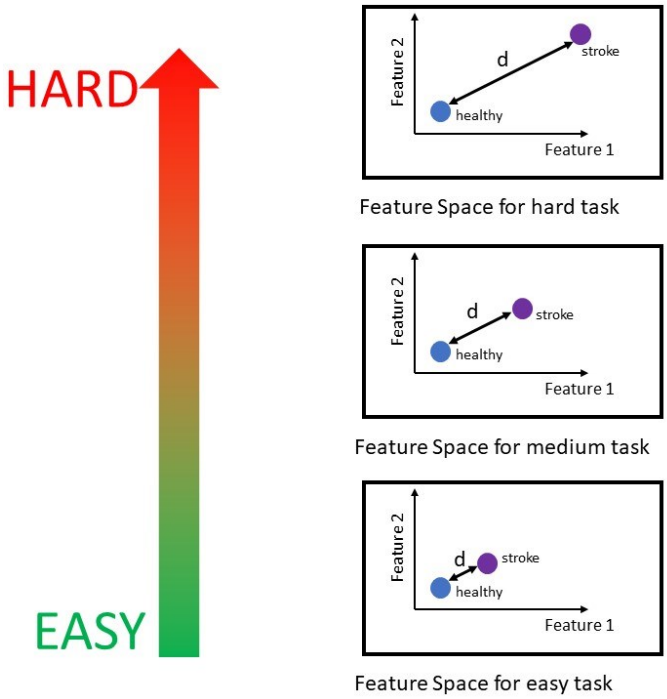



**Figure 64: Example showing how the euclidean distance (d) will be used to rank tasks. The easiest task will have the smallest distance while the hardest task will have the furthest.**

To get a perspective of how an individual patient might perform as compared to the overall stroke population, the task hierarchy was created for the general stroke population (cluster created for all stroke participants) and for an individual stroke patient. The next sections present the calculation of these task hierarchies.

### 6.2.1 Task Hierarchy of Stroke Population

To calculate the task difficulty for the stroke population, the stroke cluster needed to be defined. Since some participants had trouble completing some of the tasks, yet this result did not show up in the feature space, a penalty needed to be created to shift the stroke cluster away from the healthy to show that it was a difficult task. After the feature array was populated with all of the completed tasks, the remaining empty elements from incomplete tasks were populated with the penalty term shown in equation (6-3).

$$z_j^* = \underset{j}{argmax} \left| C_i^{Healthy} - z_{i,j} \right| \tag{6-3}$$

where $z_j^*$ represents the penalty term feature. The penalty equation finds the participant *j* that is furthest away from the healthy centroid and populates the remaining empty elements with the ith feature. Note that this is a conservative estimation. It is arguable that since the patient could not complete the task that the penalty could be more severe. The stroke centroid is then calculated using equations (6-4) and (6-5).

$$C_i^{Stroke} = \frac{1}{13} \sum_{j}^{13} z_{i,j} \tag{6-4}$$

$$C^{Stroke} = [C_1^{Stroke} \quad \cdots \quad C_M^{Stroke}] \tag{6-5}$$

where the centroid is averaged over the 4 mild, 8 moderate and 1 severe participant (total of 13). The Euclidean distance between the centroids of the healthy and stroke populations for a given task was determined as shown in equation (6-6).

$$d = \sqrt{\left(C_1^{Healthy} - C_1^{Stroke}\right)^2 + \cdots + \left(C_M^{Healthy} - C_M^{Stroke}\right)^2} \tag{6-6}$$

The process shown in equations (6-4) – (6-6) was repeated for each task. For cases when they did not finish, the empty element in the feature vector was filled with the feature that was furthest away from the centroid. For cases where the task could not be classified due to no classification results, the task was thrown into an incomplete bin and labeled the hardest out of the series of tasks.

### 6.2.2 Task Hierarchy for Stroke Individual

To define the task hierarchy of an individual stroke patient, their location in the feature space was defined as shown in equations (6-7) and (6-8).

$$C_i^{part} = \frac{1}{5}\sum_{j}^{5} z_{i,j} \tag{6-7}$$

$$C^{part} = \left[C_1^{part} \quad \cdots \quad C_M^{part}\right] \tag{6-8}$$

where each element of the centroid was the *i*th feature averaged over the 5 trials. Once the centroid was defined, the Euclidean distance was calculated using equation (6-9).

$$d = \sqrt{\left(C_1^{Healthy} - C_1^{Part}\right)^2 + \cdots + \left(C_M^{Healthy} - C_M^{Part}\right)^2} \quad \textbf{(6-9)}$$

If a participant could not complete a task, it was labeled incomplete and put in a bin as the hardest task. For some of the tasks, there were cases where only a few mild participants completed a task that could not be classified from the ensemble SVM classifier due to insufficient data from the moderate/severe class. If this was the case, the task would be labeled complete, but a rank could not be assigned to it. In future studies when more data is available, these tasks will be given a rank.

## 6.3 Aim 3: Summary of Findings

---

*Measure of success: Automated goal-oriented rehabilitation development.*

---

Using the method described in Section 6.2.1, the task hierarchy for the stroke population is shown in Figure 65 along with the labelled target rig in Figure 66.

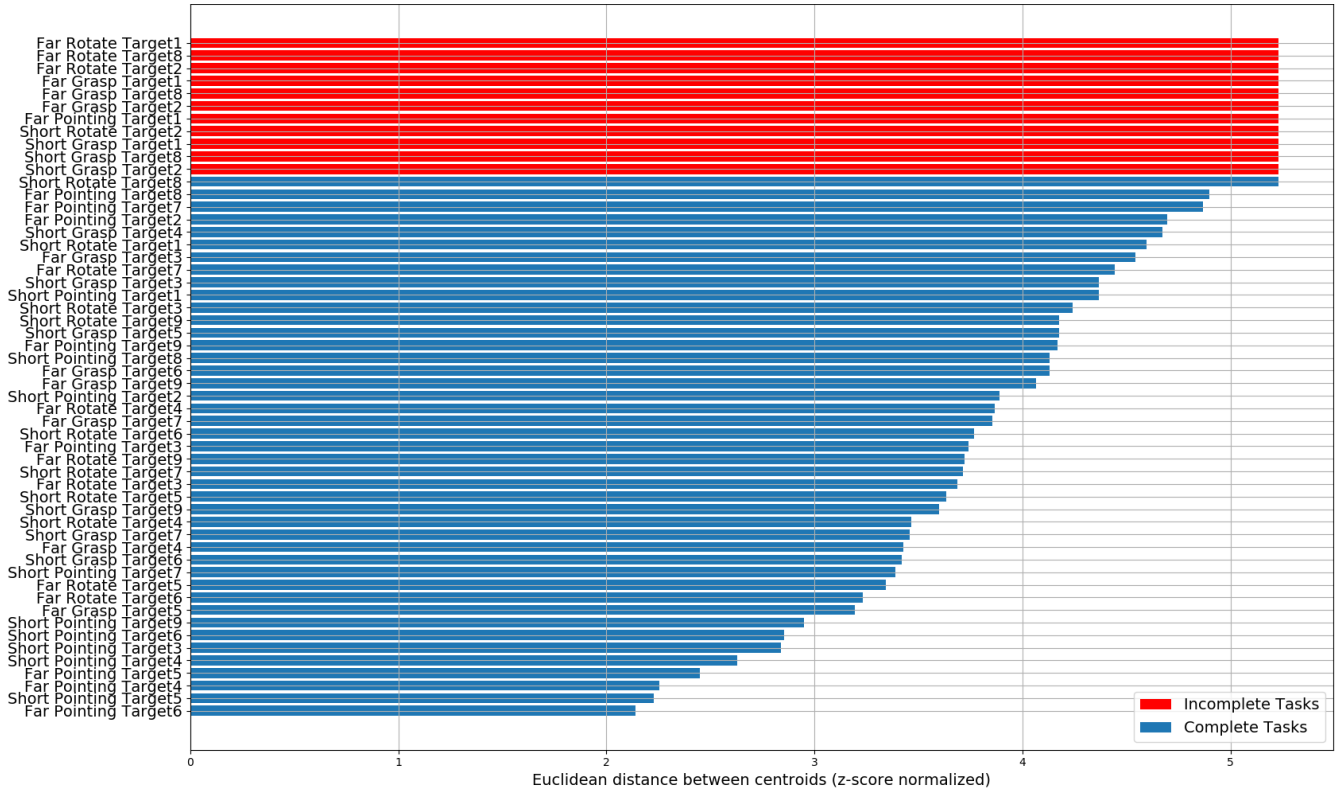



**Figure 65: The task hierarchy for the whole stroke population. The euclidean distance on the x axis is the distance away from the healthy centroid.**

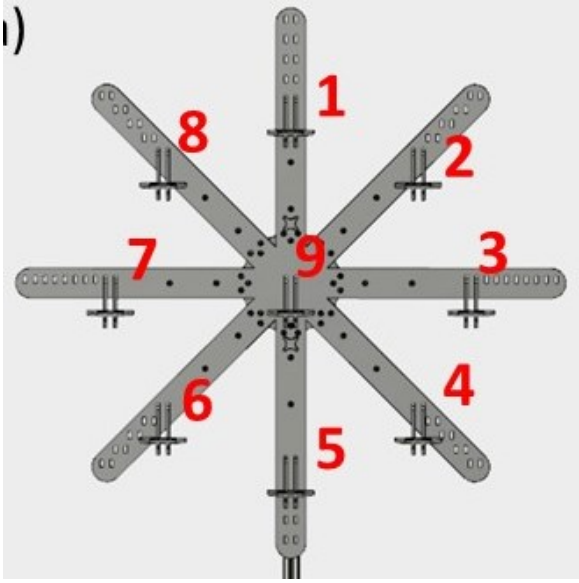



**Figure 66: CAD drawing of the target system. The red numbers represent the target numbers.**

The most difficult tasks were found to be the incomplete tasks that the moderate/severe participants could not complete. Most of these tasks include the top 3 targets (Target 1,2 and 8) during the far position (target is 80% of arm length away from chest). For the completed targets, short rotation for target 2 ended up being the most difficult with short pointing at target 5 being the easiest.

Note that the task hierarchy changes depending on the handiness of the participant. For example, Target 7 was harder for a right-handed stroke patient than a left-handed patient since it requires more coordination to reach a target across their body.

It is also important to note that, the short grasp and rotation tasks that involve more coordination within the forearm pronation and the wrist joint angles were harder than some of the

far pointing tasks. This outcome is to be expected since pointing mainly involves the coordination of the shoulder and elbow joints where the grasping tasks required additional coordination from the wrist to complete.

In some cases, a stroke participant might be able to follow the trend of their general classification population but may run into some trouble for medium level task in the middle of the task hierarchy. To see the necessity of an individualized task hierarchy, Figure 67 – Figure 69 show the hierarchies of a mild, moderate and the one severe participant in their respective orders.

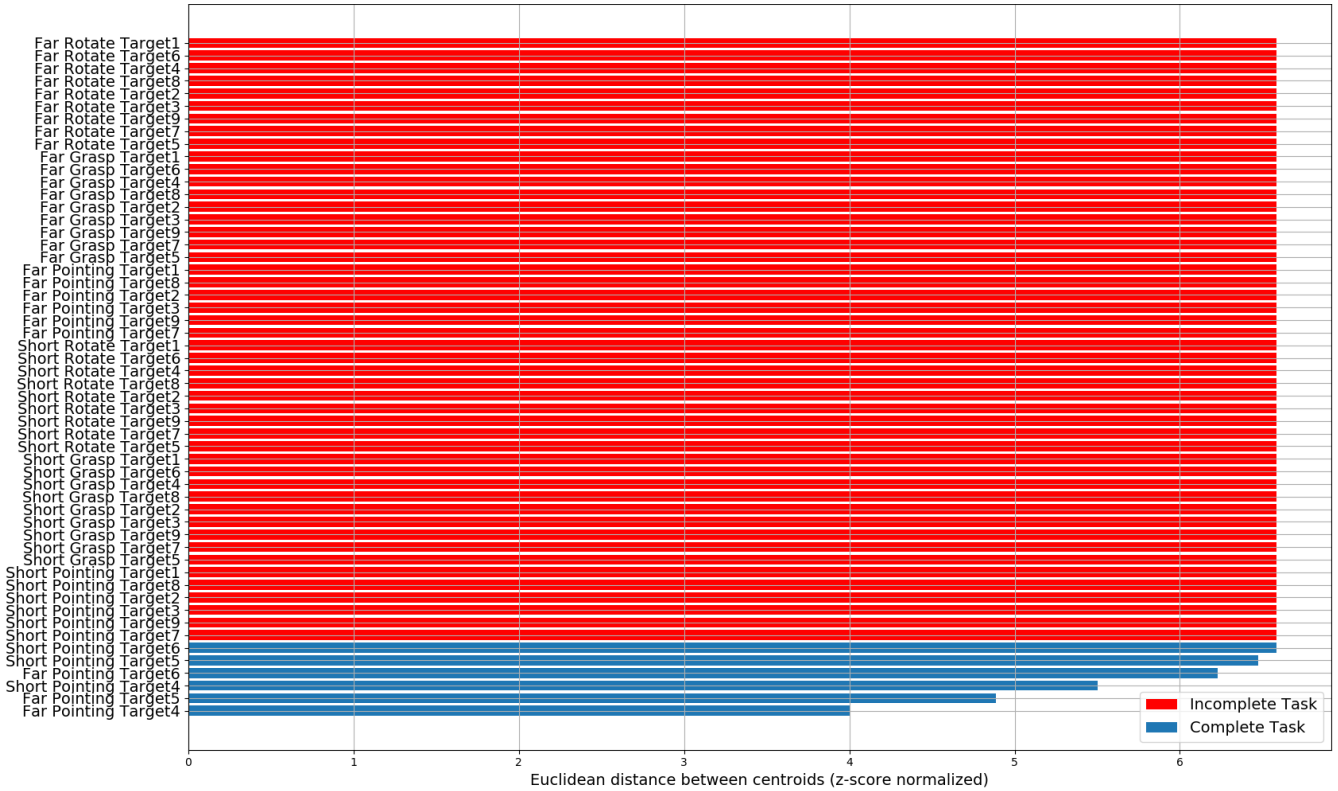



**Figure 67: Task hierarchy of severe stroke participant.**

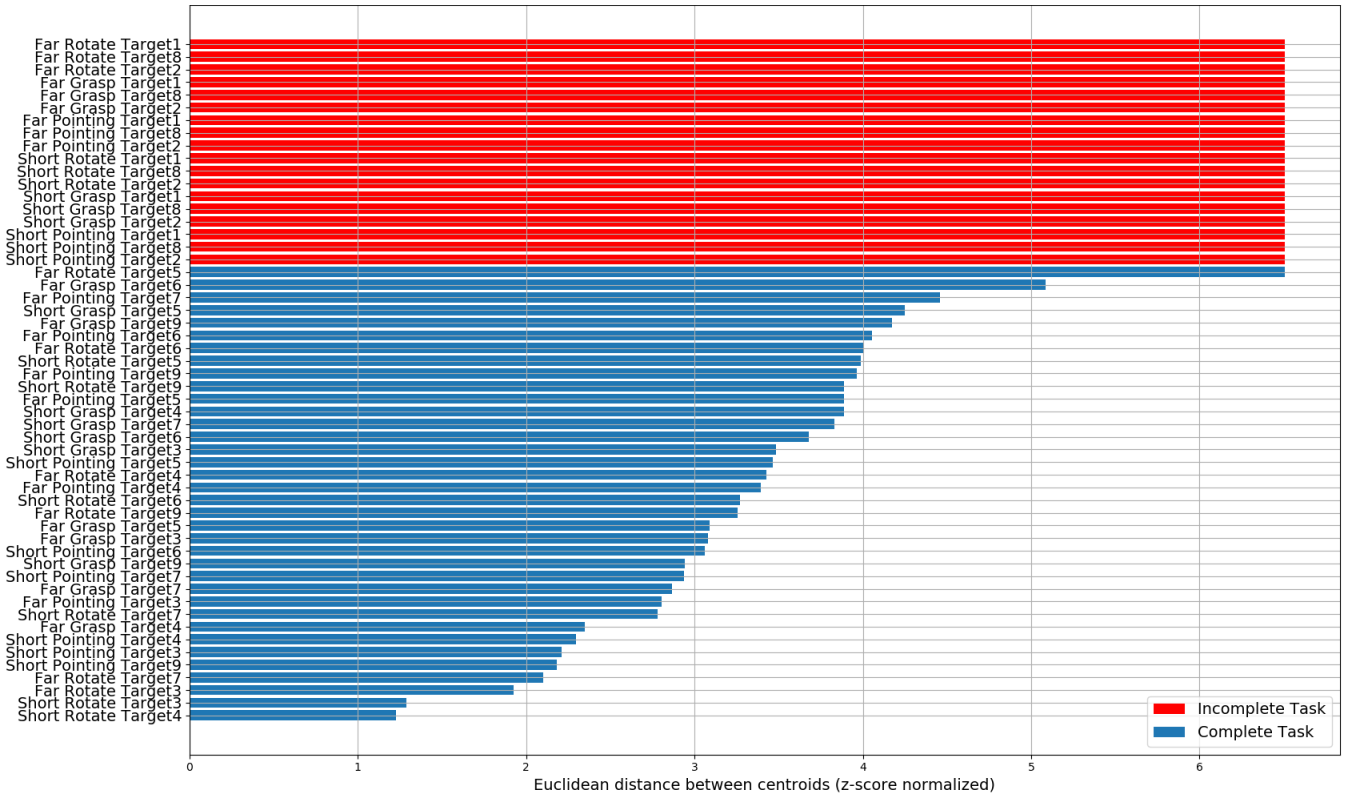



**Figure 68: Task hierarchy of moderate stroke participant.**

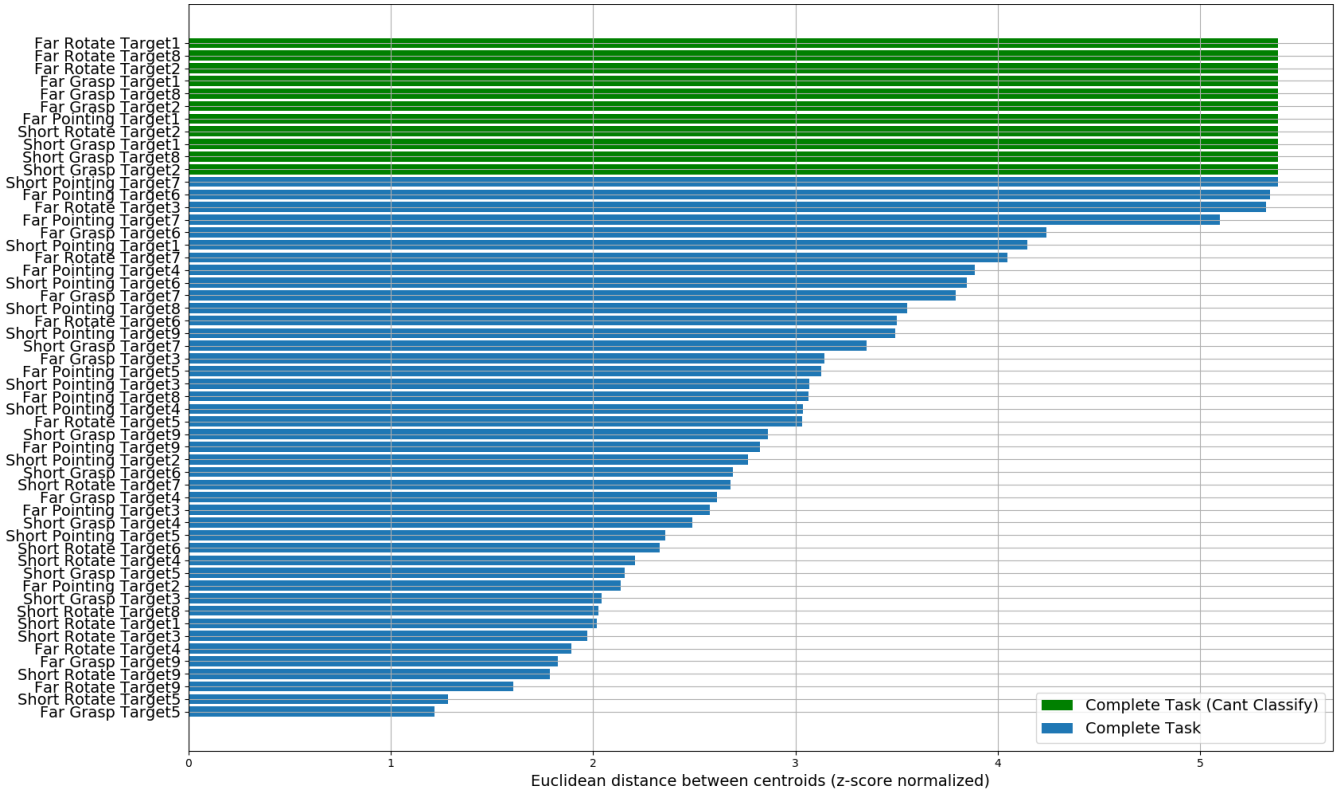



**Figure 69: Task hierarchy of mild stroke participant.**

As expected, the incomplete bins for the stroke participants increase with stroke severity. Another noticeable observation is the change in the overall scale of the Euclidean distance of the tasks between participants. That is, generally the distances increase for more severe patients. In terms of Euclidean distance, the hardest task for the mild participant is equivalent to a medium task for a moderate participant. The mild participant also showed that they were able to complete all of the tasks, but 10 tasks could not be classified due to the limited dataset.

In addition to differences across classes, the hierarchy plots also show differences within classes. For example, Figure 70 shows the task hierarchy for an additional mild stroke participant. By comparing Figure 70 with Figure 69, one can see the change in task order for patients in the same class, suggesting that patients experience the difficulty of tasks differently.

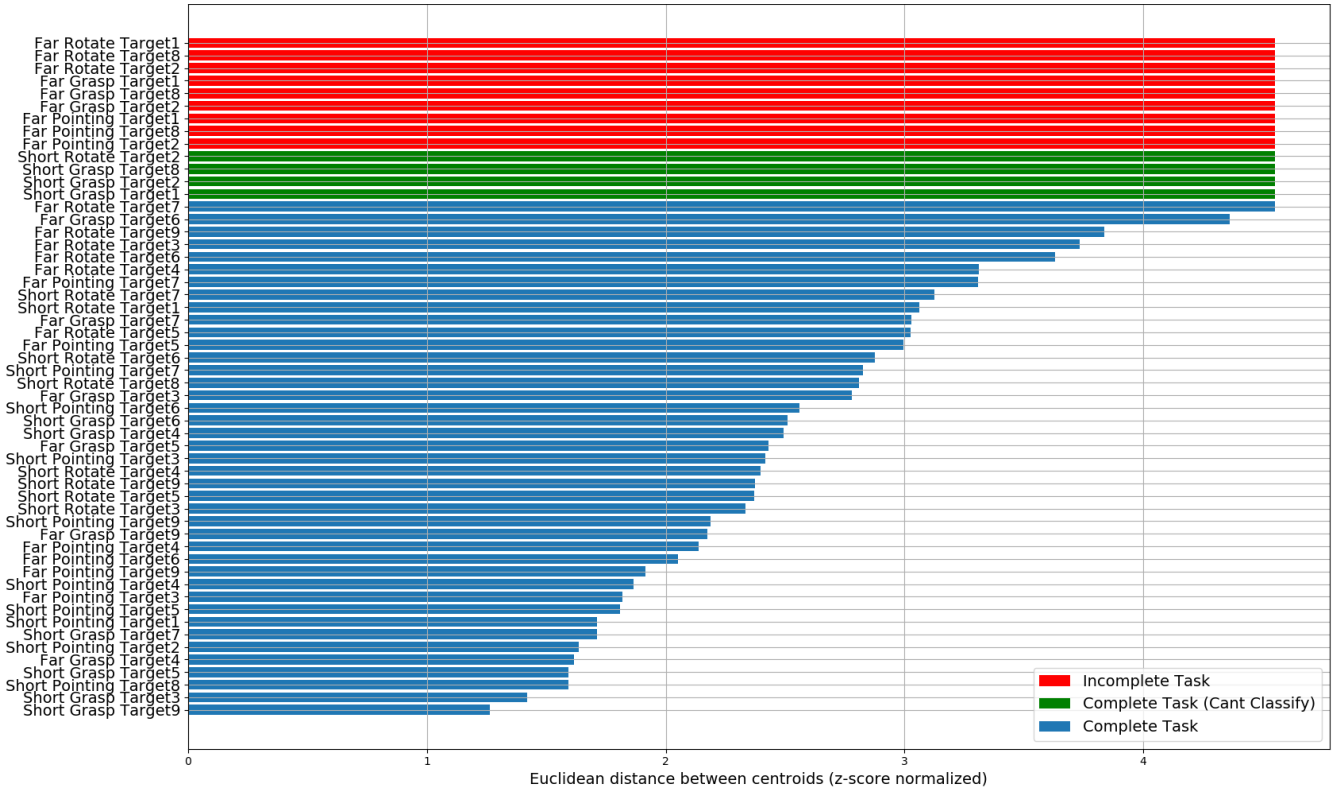



**Figure 70: A second mild stroke participant showing a different task hierarchy.**

Based on the hierarchy shown in Figure 70 for a second mild patient, this patient was unable to complete the far target 1,2, and 8 tasks, similar to the severe patient, yet some of the tasks were ranked as easier for this patient to perform than for the previous mild participant shown in Figure 69. This range of mobility for the 4 mild participants in this study may be the cause of the noise for this class shown in chapter 5.0.

It is still yet to be seen if the hierarchies calculated here are robust. For example, will a stable hierarchy be found for a given class of stroke patients stable as the tested population grows? In addition, it needs to be shown that the hierarchy for a single subject is stable over multiple tests. Nonetheless, given the patterns shown here such that the scoring trends generally match expected difficulty of the various tasks, the more severe patients show higher difficulty scores for the same tasks, and that individuals have unique hierarchies, the method holds promise for clinical use.

At the beginning of stroke rehabilitation, a clinician will prescribe these tasks that the automated system will be able to match. As a patient's FMUE score improves, the task hierarchy will likely evolve and the method shown can be used to adapt to the patient's rehabilitation regimen in real-time. Given the results shown in this chapter, it may be possible to use these methods to create an automated rehabilitation system.

## 6.4 Aim 3: Measure of Success Achievements

A personalized stroke rehabilitation system that ranks tasks from easy (motion patterns close to healthy population) to hard was created and the measure of success was met with the completion of Aim 3. In this dissertation, the research stopped with the ranking of the tasks for the

tested stroke patient population and individual stroke patients to demonstrate the method to automate goal-oriented stroke rehabilitation. To apply this task hierarchy method in a clinical setting will require testing of much larger populations. Nonetheless, the method developed in this aim is compatible with accepted motor learning methods for stroke rehabilitation. It provides a way to (1) present the patient with a set of tasks that are challenging enough to require new learning; (2) ensure that the training progresses in difficulty with real-world relevance; and (3) with real-time feedback the task can encourage repeated attempts to solve a problem instead of rote repetition. Furthermore, the method is individualized for each patient and enables automated re-evaluation at prescribed increments.

This system can demonstrate the stroke patient's improvement during their rehabilitation. As they begin to master certain tasks, the task hierarchy will evolve with mastered tasks moving towards the bottom of the hierarchy and further highlighting tasks that don't receive much attention. The expected recovery road map is assumed to be similar to the change in task hierarchy highlighted in Figure 67 – Figure 69. The patient will not only be able to feel when they regained their healthy motion patterns but also will have a quantitative result from the system when all tasks in the hierarchy have a Euclidean distance close to zero.

In order to finalize the deployment of this system, more data needs to be collected to refine the definition of healthy. With a sample size of only 10 healthy patients, there are not enough sample points to conclude that the healthy motion patterns used as a reference for the task hierarchy represent the general population. A deployable system will need at least 100 patients to be confident in the task hierarchy.

# 7.0 Conclusion

This dissertation has described a data-driven method to automate a goal-oriented stroke rehabilitation session. To establish this method, three aims were defined for this research:

1. Define a kinematic and kinetic based feature set from which to assess and predict stroke severity
2. Create a method to predict stroke severity with motion-based features using machine learning techniques.
3. Define individualized task difficulty hierarchy.

In Aim 1, the first two measures of success (1. Define kinematic and kinetic based motion features and 2. calculate motion features) were met. The joint kinematics were collected using inertial measurement units (IMUs) while the joint kinetics were calculated through an inverse dynamic model operating on the trajectories formed from the IMU data. After receiving the kinematics and kinetic readings, they were summarized in scalar values using statistical functions (mean, variance and max) and frequency domain transformations using discrete Fourier transform (DFT). Through the methods described in sections 4.5.1 and 4.5.2, the total list of motion features was defined and later refined to the top features used to predict stroke severity (motion features list is shown in Appendix B).

The last measure of success for Aim 1 (3. IMU validation) was set that the IMU joint kinematic readings had to have a cross-correlation between 0.8 – 1 with the gold standard optical tracking system (Optitrack, NaturalPoint, Inc.). This measure was met consistently for the torso and shoulder angles (i.e. torso bending, torso bending, shoulder flexion, shoulder abduction, and shoulder rotation) but torso flexion, elbow flexion, forearm pronation and wrist angles (wrist

flexion and wrist deviation) varied throughout various tasks and trials. This disagreement between these two systems originates from the different methods used to define the body coordinate system for each system. The body coordinate system definitions need to be revisited for each system to guarantee a cross-correlation of at least 0.8 for all angles. It is important to note that even though the IMU readings did not agree with the gold standard, it did not deter from the overarching goal for the IMU to predict stroke severity. For this reason, the research continued to Aim 2.

The measure of success for Aim 2 was to obtain a stroke severity prediction F1 score of at least 0.95 for each task. This measure of success was not met mainly due to the limited amount of data provided in the study. The dataset had a large disparity in sample sizes with 14 healthy, 4 mild and 9 moderate/severe participants with each participant completing 5 trials (26 participants × 5 trials=504 sample points). Unfortunately, the dataset was further reduced to 26 sample points due to redundant information between trials (only 1 trial could be used per subject in the classifier) and moderate/severe patients' inability to complete some tasks. The Synthetic Minority Oversampling Technique (SMOTE) was used to reduce the imbalance between the healthy and mild stroke severity dataset. The imbalanced caused the performance of the classifier (measured using the F1 score) to fluctuate depending on the number of stroke patients used in the training dataset. In future work, the machine learning method defined in this dissertation can show improved performance with a large patient population (+100 patients). Although the measure of success was not met, the improved performance of the stroke severity prediction as the task became more difficult for stroke participants proved to be useful for Aim 3.

Aim 3's measure of success to develop an automated goal-oriented stroke rehabilitation system that was met. This system was built by creating a task difficulty hierarchy using the top features defined in Aim 2. The two distinguishing factors that influenced the hierarchy were the

stroke severity and the handedness of the subject. The task hierarchy follows the goal-oriented motor learning method and provides a way to (1) present the patient with a set of tasks that are challenging enough to require new learning; (2) ensure that the training progresses in difficulty with real-world relevance; and (3) with real-time feedback the task can encourage repeated attempts to solve a problem instead of rote repetition. Furthermore, the method is individualized for each patient and enables automated re-evaluation at prescribed increments. With the limited dataset, it is tough to prove that this hierarchy will remain true when more data is introduced to the system but the method to obtain the hierarchy is valid.

## 7.1 Summary of Contributions

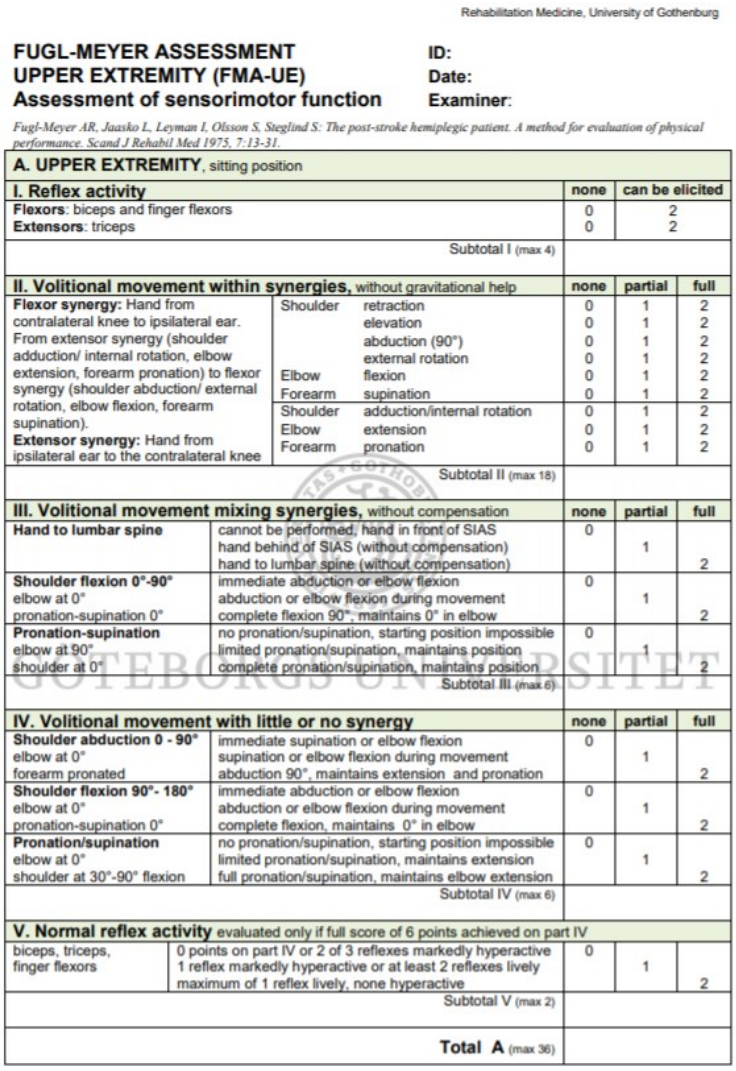
A summary of contributions towards the areas of machine learning and occupational therapy are described below:

1. Kinematic and kinetic based motion features have been extracted from the portable IMU motion capture system that can predict stroke severity. The work described in this dissertation also provides a baseline to extract these motion features when additional participants are evaluated in future studies.
2. A method to select the optimal motion features (final features) and predict stroke severity of a participant that matches a professional clinical evaluation has been established. Even though the test set was small, a key set of features was identified that were consistently selected as important in the classification process.
3. A method to develop an automated personalized task-oriented stroke rehabilitation system has been defined. With further development of this system made through

collecting more data, a deployable version of the system can help stroke participants continue to set achievable task goals at home and accelerate their recovery time.

## Appendix A : Fugl-Meyer Upper Extremity Assessment

The official version of the Fugl-Meyer Upper Extremity Assessment from Fugl-Meyer (Fugl-Meyer A.R., 1975) is provided in Figure 71 and Figure 72.



Rehabilitation Medicine, University of Gothenburg

**FUGL-MEYER ASSESSMENT** **ID:**
**UPPER EXTREMITY (FMA-UE)** **Date:**
**Assessment of sensorimotor function** **Examiner:**

*Fugl-Meyer AR, Jaasko L, Leyman I, Olsson S, Steglind S: The post-stroke hemiplegic patient. A method for evaluation of physical performance. Scand J Rehabil Med 1975, 7:13-31.*

| **A. UPPER EXTREMITY**, sitting position | | | | | |
|---|---|---|---|---|---|
| **I. Reflex activity** | | | **none** | **can be elicited** | |
| **Flexors**: biceps and finger flexors<br>**Extensors**: triceps | | | 0<br>0 | 2<br>2 | |
| Subtotal I (max 4) | | | | | |
| **II. Volitional movement within synergies,** without gravitational help | | | **none** | **partial** | **full** |
| **Flexor synergy:** Hand from contralateral knee to ipsilateral ear. From extensor synergy (shoulder adduction/ internal rotation, elbow extension, forearm pronation) to flexor synergy (shoulder abduction/ external rotation, elbow flexion, forearm supination).<br>**Extensor synergy:** Hand from ipsilateral ear to the contralateral knee | Shoulder<br><br><br><br>Elbow<br>Forearm | retraction<br>elevation<br>abduction (90°)<br>external rotation<br>flexion<br>supination | 0<br>0<br>0<br>0<br>0<br>0 | 1<br>1<br>1<br>1<br>1<br>1 | 2<br>2<br>2<br>2<br>2<br>2 |
| | Shoulder<br>Elbow<br>Forearm | adduction/internal rotation<br>extension<br>pronation | 0<br>0<br>0 | 1<br>1<br>1 | 2<br>2<br>2 |
| Subtotal II (max 18) | | | | | |
| **III. Volitional movement mixing synergies,** without compensation | | | **none** | **partial** | **full** |
| **Hand to lumbar spine** | cannot be performed, hand in front of SIAS<br>hand behind of SIAS (without compensation)<br>hand to lumbar spine (without compensation) | | 0 | 1 | 2 |
| **Shoulder flexion 0°-90°**<br>elbow at 0°<br>pronation-supination 0° | immediate abduction or elbow flexion<br>abduction or elbow flexion during movement<br>complete flexion 90°, maintains 0° in elbow | | 0 | 1 | 2 |
| **Pronation-supination**<br>elbow at 90°<br>shoulder at 0° | no pronation/supination, starting position impossible<br>limited pronation/supination, maintains position<br>complete pronation/supination, maintains position | | 0 | 1 | 2 |
| Subtotal III (max 6) | | | | | |
| **IV. Volitional movement with little or no synergy** | | | **none** | **partial** | **full** |
| **Shoulder abduction 0 - 90°**<br>elbow at 0°<br>forearm pronated | immediate supination or elbow flexion<br>supination or elbow flexion during movement<br>abduction 90°, maintains extension and pronation | | 0 | 1 | 2 |
| **Shoulder flexion 90°- 180°**<br>elbow at 0°<br>pronation-supination 0° | immediate abduction or elbow flexion<br>abduction or elbow flexion during movement<br>complete flexion, maintains 0° in elbow | | 0 | 1 | 2 |
| **Pronation/supination**<br>elbow at 0°<br>shoulder at 30°-90° flexion | no pronation/supination, starting position impossible<br>limited pronation/supination, maintains extension<br>full pronation/supination, maintains elbow extension | | 0 | 1 | 2 |
| Subtotal IV (max 6) | | | | | |
| **V. Normal reflex activity** evaluated only if full score of 6 points achieved on part IV | | | | | |
| biceps, triceps,<br>finger flexors | 0 points on part IV or 2 of 3 reflexes markedly hyperactive<br>1 reflex markedly hyperactive or at least 2 reflexes lively<br>maximum of 1 reflex lively, none hyperactive | | 0 | 1 | 2 |
| Subtotal V (max 2) | | | | | |
| **Total A** (max 36) | | | | | |

**Figure 71: The first page of the FMUE Assessment.**

| **B. WRIST** support may be provided at the elbow to take or hold the position, no support at wrist, check the passive range of motion prior testing | | none | partial | full |
|---|---|---|---|---|
| **Stability at 15° dorsiflexion**<br>elbow at 90°, forearm pronated<br>shoulder at 0° | less than 15° active dorsiflexion<br>dorsiflexion 15°, no resistance is taken<br>maintains position against resistance | 0 | 1 | 2 |
| **Repeated dorsifexion / volar flexion**<br>elbow at 90°, forearm pronated<br>shoulder at 0°, slight finger flexion | cannot perform volitionally<br>limited active range of motion<br>full active range of motion, smoothly | 0 | 1 | 2 |
| **Stability at 15° dorsiflexion**<br>elbow at 0°, forearm pronated<br>slight shoulder flexion/abduction | less than 15° active dorsiflexion<br>dorsiflexion 15°, no resistance is taken<br>maintains position against resistance | 0 | 1 | 2 |
| **Repeated dorsifexion / volar flexion**<br>elbow at 0°, forearm pronated<br>slight shoulder flexion/abduction | cannot perform volitionally<br>limited active range of motion<br>full active range of motion, smoothly | 0 | 1 | 2 |
| **Circumduction** | cannot perform volitionally<br>jerky movement or incomplete<br>complete and smooth circumduction | 0 | 1 | 2 |
| | **Total B** (max 10) | | | |

| **C. HAND** support may be provided at the elbow to keep 90° flexion, no support at the wrist, compare with unaffected hand, the objects are interposed, active grasp | | none | partial | full |
|---|---|---|---|---|
| **Mass flexion**<br>from full active or passive extension | | 0 | 1 | 2 |
| **Mass extension**<br>from full active or passive flexion | | 0 | 1 | 2 |
| **GRASP** | | | | |
| **A – flexion in PIP and DIP** (digits II-V)<br>**extension in MCP II-V** | cannot be performed<br>can hold position but weak<br>maintains position against resistance | 0 | 1 | 2 |
| **B – thumb adduction**<br>1-st CMC, MCP, IP at 0°, scrap of paper between thumb and 2-nd MCP joint | cannot be performed<br>can hold paper but not against tug<br>can hold paper against a tug | 0 | 1 | 2 |
| **C - opposition** pulpa of the thumb against the pulpa of 2-nd finger, pencil, tug upward | cannot be performed<br>can hold pencil but not against tug<br>can hold pencil against a tug | 0 | 1 | 2 |
| **D – cylinder grip**<br>cylinder shaped object (small can)<br>tug upward, opposition in digits I and II | cannot be performed<br>can hold cylinder but not against tug<br>can hold cylinder against a tug | 0 | 1 | 2 |
| **E – spherical grip**<br>fingers in abduction/flexion, thumb opposed, tennis ball | cannot be performed<br>can hold ball but not against tug<br>can hold ball against a tug | 0 | 1 | 2 |
| | **Total C** (max 14) | | | |

| **D. COORDINATION/SPEED** after one trial with both arms, blind-folded, tip of the index finger from knee to nose, 5 times as fast as possible | | marked | slight | none |
|---|---|---|---|---|
| **Tremor** | | 0 | 1 | 2 |
| **Dysmetria** | pronounced or unsystematic<br>slight and systematic<br>no dysmetria | 0 | 1 | 2 |
| | | **> 5s** | **2 - 5s** | **< 1s** |
| **Time** | more than 5 seconds slower than unaffected side<br>2-5 seconds slower than unaffected side<br>maximum difference of1 second between sides | 0 | 1 | 2 |
| | **Total D** (max 6) | | | |

| **TOTAL A-D** (max 66) | |
|---|---|



**Figure 72: The second page for the FMUE Assessment.**

## Appendix B : Final List of Motion Features

The total list of motion features calculated in this study are shown in Table 4. The first 154 features represent the Candidate features used as a pool from which to select the top features that predict stroke severity as shown in section 5.6 (top features for a given task are the five most frequently selected features using the feature selection process described in chapter 5.0). The process of narrowing down the motion features is shown in the Venn Diagram in Figure 73.

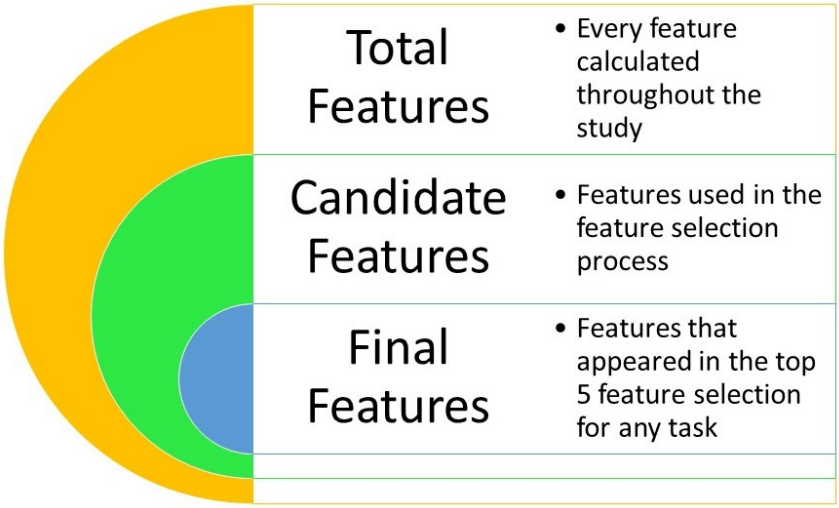



**Figure 73: The Venn diagram shows how the features were narrowed down to the top features that can classify stroke severity.**

The remaining features were once considered for the study but removed due to never being selected in the top features to predict stroke severity. To provide further detail for the features, Appendix C list the equations used to calculate each feature.

**Table 4: List of motion features used to analyze stroke severity.**

| Candidate Features | | |
|---|---|---|
| **#** | **Feature** | **Tasks that selected the candidate feature as a top 5 feature** |
| 1 | Average chest displacement | Short Pointing Target 1 – 9<br>Short Grasp Target 3 – 7,9<br>Short Rotate Target 1,3 – 9<br>Far Pointing Target 2 – 9<br>Far Grasp Target 3 – 7,9<br>Far Rotate Target 3 – 7,9 |
| 2 | Variance of torso lateral bending | Short Pointing Target 1 – 9<br>Short Grasp Target 3 – 7,9<br>Short Rotate Target 1,3 – 9<br>Far Pointing Target 2 – 9<br>Far Grasp Target 3 – 7,9<br>Far Rotate Target 3 – 7,9 |

**Table 4 (continued)**

| | | |
|---|---|---|
| 3 | Variance of torso axial rotation | Short Pointing Target 5,8<br>Short Grasp Target 4<br>Short Rotate Target 5,6,8,9<br>Far Rotate Target 5,6,7,9 |
| 4 | Variance of torso flexion | Short Pointing Target 4<br>Far Pointing Target 9<br>Far Rotate Target 4 |
| 5 | Variance of shoulder abduction | Short Pointing Target 6,7<br>Short Rotate Target 7<br>Far Pointing Target 6. |
| 6 | Variance of shoulder rotation | Short Pointing Target 5<br>Short Grasp Target 3<br>Far Grasp Target 9 |
| 7 | Variance of shoulder flexion | Far Pointing Target 7 |
| 8 | Variance of elbow flexion | |
| 9 | Variance of wrist flexion | |
| 10 | Variance of forearm pronation | |
| 11 | Variance of wrist deviation | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 12 | Zero crossing of torso lateral bending velocity | Short Pointing Target 1 – 9<br>Short Grasp Target 3 – 7,9<br>Short Rotate Target 1,3 – 9<br>Far Pointing Target 2 – 9<br>Far Grasp Target 3 – 7,9<br>Far Rotate Target 3 – 7,9 |
| 13 | Zero crossing of torso axial rotation velocity | |
| 14 | Zero crossing of torso flexion velocity | |
| 15 | Zero crossing of torso lateral bending acceleration | Short Pointing Target 1 – 9<br>Short Grasp Target 3,5,6<br>Far Pointing Target 2 – 9<br>Far Grasp Target 3,5,6<br>Far Rotate Target 3 – 6,7,9 |
| 16 | Zero crossing of torso axial rotation acceleration | |
| 17 | Zero crossing of torso flexion acceleration | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 18 | Zero crossing of shoulder abduction velocity | Short Grasp Target 9 |
| 19 | Zero crossing of shoulder rotation velocity | Far Pointing Target 7<br>Far Grasp Target 6,9 |
| 20 | Zero crossing of shoulder flexion velocity | |
| 21 | Zero crossing of shoulder abduction acceleration | |
| 22 | Zero crossing of shoulder rotation acceleration | Short Grasp Target 4<br>Far Grasp Target 6 |
| 23 | Zero crossing of shoulder flexion acceleration | |
| 24 | Zero crossing of elbow flexion velocity | Far Rotate Target 4 |
| 25 | Zero crossing of elbow flexion acceleration | Far Rotate Target 4 |
| 26 | Zero crossing of wrist flexion velocity | |
| 27 | Zero crossing of forearm pronation velocity | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 28 | Zero crossing of wrist deviation velocity | |
| 29 | Zero crossing of wrist flexion acceleration | |
| 30 | Zero crossing of forearm pronation acceleration | |
| 31 | Zero crossing of wrist deviation acceleration | |
| 32 | Max amplitude of torso lateral bending velocity (Freq. domain) | Short Pointing Target 1,8<br>Short Grasp Target 3 – 4<br>Short Rotate Target 1,3,4<br>Far Pointing Target 2 – 4,6<br>Far Grasp Target 3.4.7 – 9<br>Far Rotate Target 5 – 7 |
| 33 | Max amplitude of torso axial rotation velocity (Freq. domain) | Short Pointing Target 4 |
| 34 | Max amplitude of torso flexion velocity (Freq. domain) | Short Grasp Target 7 |

**Table 4 (continued)**

| | | |
|---|---|---|
| 35 | 2nd Max amplitude of torso lateral bending velocity (Freq. domain) | Short Pointing Target 6,7<br>Short Grasp Target 4,5<br>Short Rotate Target 3,4,7<br>Far Grasp Target 3,5 – 7<br>Far Rotate Target 5 – 7 |
| 36 | 2nd Max amplitude of torso axial rotation velocity (Freq. domain) | Short Grasp Target 4<br>Short Rotate Target 6<br>Far Grasp Target 4 |
| 37 | 2nd Max amplitude of torso flexion velocity (Freq. domain) | Short Pointing Target 6<br>Short Grasp Target 5 – 7,9<br>Far Pointing Target 6<br>Far Grasp Target 6<br>Far Rotate Target 5 |
| 38 | Freq of Max amplitude of torso lateral bending velocity (Freq. domain) | Short Pointing Target 1 – 5,8<br>Short Grasp Target 3 – 7,9<br>Short Rotate Target 1,3 – 9<br>Far Pointing Target 5 – 9<br>Far Grasp Target 3 – 7<br>Far Rotate Target 3,5,7 |

**Table 4 (continued)**

| | | |
|---|---|---|
| 39 | Freq of Max amplitude of torso axial rotation velocity (Freq. domain) | |
| 40 | Freq of Max amplitude of torso flexion velocity (Freq. domain) | Far Grasp Target 7 |
| 41 | Freq of 2nd Max amplitude of torso lateral bending velocity (Freq. domain) | Short Grasp Target 3,4<br>Short Rotate Target 9<br>Far Grasp Target 5,9 |
| 42 | Freq of 2nd Max amplitude of torso axial rotation velocity (Freq. domain) | Far Grasp Target 4 |
| 43 | Freq of 2nd Max amplitude of torso flexion velocity (Freq. domain) | Short Pointing Target 6,7<br>Short Grasp Target 6,7<br>Short Rotate Target 4,6 |
| 44 | Max amplitude of shoulder abduction velocity (Freq. domain) | Short Grasp Target 9 |

**Table 4 (continued)**

| | | |
|---|---|---|
| 45 | Max amplitude of shoulder rotation velocity (Freq. domain) | Short Pointing Target 3,9<br>Short Grasp Target 4<br>Far Pointing Target 4,7 |
| 46 | Max amplitude of shoulder flexion velocity (Freq. domain) | |
| 47 | 2nd Max amplitude of shoulder abduction velocity (Freq. domain) | Short Rotate Target 9<br>Far Pointing Target 9<br>Far Grasp Target 9<br>Far Rotate Target 9 |
| 48 | 2nd Max amplitude of shoulder rotation velocity (Freq. domain) | Short Pointing Target 6, 9<br>Far Pointing Target 4 |
| 49 | 2nd Max amplitude of shoulder flexion velocity (Freq. domain) | |
| 50 | Freq of Max amplitude of shoulder abduction velocity (Freq. domain) | Short Grasp Target 7,9<br>Far Grasp Target 9<br>Far Rotate Target 9 |
| 51 | Freq of Max amplitude of shoulder rotation velocity (Freq. domain) | Short Pointing Target 7,9<br>Far Pointing Target 4,6 |

**Table 4 (continued)**

| | | |
|---|---|---|
| 52 | Freq of Max amplitude of shoulder flexion velocity (Freq. domain) | Far Pointing Target 4 |
| 53 | Freq of 2nd Max amplitude of shoulder abduction velocity (Freq. domain) | Far Rotate Target 9 |
| 54 | Freq of 2nd Max amplitude of shoulder rotation velocity (Freq. domain) | Short Pointing Target 7,9<br>Far Pointing Target 6,9<br>Far Rotate Target 4 |
| 55 | Freq of 2nd Max amplitude of shoulder flexion velocity (Freq. domain) | |
| 56 | Max amplitude of elbow flexion velocity (Freq. domain) | Far Pointing Target 5 |
| 57 | 2nd Max amplitude of elbow flexion velocity (Freq. domain) | |
| 58 | Freq of Max amplitude of elbow flexion velocity (Freq. domain) | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 59 | Freq of 2nd Max amplitude of elbow flexion velocity (Freq. domain) | |
| 60 | Max amplitude of wrist deviation velocity (Freq. domain) | |
| 61 | Max amplitude of forearm pronation velocity (Freq. domain) | |
| 62 | Max amplitude of wrist flexion velocity (Freq. domain) | |
| 63 | 2nd Max amplitude of wrist deviation velocity (Freq. domain) | |
| 64 | 2nd Max amplitude of forearm pronation velocity (Freq. domain) | |
| 65 | 2nd Max amplitude of wrist flexion velocity (Freq. domain) | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 66 | Freq of Max amplitude of wrist deviation velocity (Freq. domain) | |
| 67 | Freq of Max amplitude of forearm pronation velocity (Freq. domain) | Short Pointing Target 4 |
| 68 | Freq of Max amplitude of wrist flexion velocity (Freq. domain) | |
| 69 | Freq of 2nd Max amplitude of wrist deviation velocity (Freq. domain) | |
| 70 | Freq of 2nd Max amplitude of forearm pronation velocity (Freq. domain) | |
| 71 | Freq of 2nd Max amplitude of wrist flexion velocity (Freq. domain) | |
| 72 | Spearman correlation between torso and shoulder flexion | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 73 | Spearman correlation between shoulder and elbow flexion | |
| 74 | Spearman correlation between elbow and wrist flexion | |
| 75 | Torso lateral bending maximum amplitude focal point toque (Freq. domain) | |
| 76 | Torso axial rotation maximum amplitude focal point toque (Freq. domain) | |
| 77 | Torso flexion maximum amplitude focal point toque (Freq. domain) | |
| 78 | Torso lateral bending maximum amplitude interaction torque (Freq. domain) | |
| 79 | Torso axial rotation maximum amplitude interaction torque (Freq. domain) | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 80 | Torso flexion maximum amplitude interaction torque (Freq. domain) | |
| 81 | Torso lateral bending maximum amplitude gravity torque (Freq. domain) | |
| 82 | Torso axial rot maximum amplitude gravity torque (Freq. domain) | |
| 83 | Torso flexion maximum amplitude gravity torque (Freq. domain) | |
| 84 | Torso lateral bending maximum amplitude muscle torque (Freq. domain) | |
| 85 | Torso axial rotation maximum amplitude muscle torque (Freq. domain) | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 86 | Torso flexion maximum amplitude muscle torque (Freq. domain) | |
| 87 | Torso lateral bending dominant frequency focal point torque | |
| 88 | Torso axial rotation dominant frequency focal point torque | |
| 89 | Torso flexion dominant frequency focal point torque | |
| 90 | Torso lateral bending dominant frequency interaction torque | |
| 91 | Torso axial rot dominant frequency interaction torque | |
| 92 | Torso flexion dominant frequency interaction torque | |
| 93 | Torso lateral bending dominant frequency gravity torque | |
| 94 | Torso axial rot dominant frequency gravity torque | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 95 | Torso flexion dominant frequency gravity torque | |
| 96 | Torso lateral bending dominant frequency muscle torque | |
| 97 | Torso axial rot dominant frequency muscle torque | |
| 98 | Torso flexion dominant frequency muscle torque | |
| 99 | Shoulder abduction maximum amplitude focal point torque (Freq. domain) | |
| 100 | Shoulder rotation maximum amplitude focal point torque (Freq. domain) | |
| 101 | Shoulder flexion maximum amplitude focal point torque (Freq. domain) | |
| 102 | Shoulder abduction maximum amplitude interaction torque (Freq. domain) | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 103 | Shoulder rotation maximum amplitude interaction torque (Freq. domain) | |
| 104 | Shoulder flexion maximum amplitude interaction torque (Freq. domain) | |
| 105 | Shoulder abduction maximum amplitude gravity torque (Freq. domain) | |
| 106 | Shoulder rotation maximum amplitude gravity torque (Freq. domain) | |
| 107 | Shoulder flexion maximum amplitude gravity torque (Freq. domain) | |
| 108 | Shoulder abduction maximum amplitude muscle torque (Freq. domain) | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 109 | Shoulder abduction maximum amplitude muscle torque (Freq. domain) | |
| 110 | Shoulder flexion maximum amplitude muscle torque (Freq. domain) | |
| 111 | Shoulder abduction dominant frequency focal point torque | |
| 112 | Shoulder abduction dominant frequency focal point torque | |
| 113 | Shoulder flexion dominant frequency focal point torque | |
| 114 | Shoulder abduction dominant frequency interaction torque | |
| 115 | Shoulder rotation dominant frequency interaction torque | |
| 116 | Shoulder flexion dominant frequency interaction torque | |
| 117 | Shoulder abduction dominant frequency GT | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 118 | Shoulder rotation dominant frequency gravity torque | |
| 119 | Shoulder flexion dominant frequency gravity torque | |
| 120 | Shoulder abduction dominant frequency muscle torque | |
| 121 | Shoulder rotation dominant frequency muscle torque | |
| 122 | Shoulder flexion dominant frequency muscle torque | |
| 123 | Elbow Flexion maximum amplitude focal point torque (Freq. domain) | |
| 124 | Elbow Flexion maximum amplitude interaction torque (Freq. domain) | |
| 125 | Elbow Flexion maximum amplitude gravity torque (Freq. domain) | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 126 | Elbow Flexion maximum amplitude muscle torque (Freq. domain) | |
| 127 | Elbow flexion dominant frequency focal point torque | |
| 128 | Elbow flexion dominant frequency interaction torque | |
| 129 | Elbow flexion dominant frequency gravity torque | |
| 130 | Elbow flexion dominant frequency muscle torque | |
| 131 | Wrist flexion maximum amplitude focal point torque (Freq. domain) | |
| 132 | Forearm pronation maximum amplitude focal point torque (Freq. domain) | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 133 | Wrist deviation maximum amplitude focal point torque (Freq. domain) | |
| 134 | Wrist flexion maximum amplitude interaction torque (Freq. domain) | |
| 135 | Forearm pronation maximum amplitude interaction torque (Freq. domain) | |
| 136 | Wrist deviation maximum amplitude interaction torque (Freq. domain) | |
| 137 | Wrist flexion maximum amplitude gravity torque (Freq. domain) | |
| 138 | Forearm pronation maximum amplitude gravity torque (Freq. domain) | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 139 | Wrist deviation maximum amplitude gravity torque (Freq. domain) | |
| 140 | Wrist flexion maximum amplitude muscle torque (Freq. domain) | |
| 141 | Forearm pronation maximum amplitude muscle torque (Freq. domain) | |
| 142 | Wrist deviation maximum amplitude muscle torque (Freq. domain) | |
| 143 | Wrist flexion dominant frequency focal point torque | |
| 144 | Forearm pronation dominant frequency focal point torque | |
| 145 | Wrist deviation dominant frequency focal point torque | |

**Table 4 (continued)**

| | | |
|---|---|---|
| 146 | Wrist flexion dominant frequency interaction torque | |
| 147 | Forearm pronation dominant frequency interaction torque | |
| 148 | Wrist deviation dominant frequency interaction torque | |
| 149 | Wrist flexion dominant frequency gravity torque | |
| 150 | Forearm pronation dominant frequency gravity torque | |
| 151 | Wrist deviation dominant frequency gravity torque | |
| 152 | Wrist flexion dominant frequency muscle torque | |
| 153 | Forearm pronation dominant frequency muscle torque | |
| 154 | Wrist deviation dominant frequency muscle torque | |

**Table 4 (continued)**

| | Remaining Features Previously Considered |
|---|---|
| 155 | Max Torso lateral bending |
| 156 | Max Torso axial rotation |
| 157 | Max torso flexion |
| 158 | Max torso lateral bending velocity |
| 159 | Max torso axial rotation |
| 160 | Max torso flexion velocity |
| 161 | Max torso lateral bending acceleration |
| 162 | Max torso axial rotation acceleration |
| 163 | Max torso flexion acceleration |
| 164 | Max torso lateral bending jerk |
| 165 | Max torso axial rotation jerk |
| 166 | Max torso flexion jerk |
| 167 | Max shoulder abduction |
| 168 | Max shoulder rotation |
| 169 | Max shoulder flexion |
| 170 | Max shoulder abduction velocity |
| 171 | Max shoulder rotation velocity |
| 172 | Max shoulder flexion velocity |
| 173 | Max shoulder abduction acceleration |

**Table 4 (continued)**

| | |
|---|---|
| 174 | Max shoulder rotation acceleration |
| 175 | Max shoulder flexion acceleration |
| 176 | Max shoulder abduction jerk |
| 177 | Max shoulder rotation jerk |
| 178 | Max shoulder flexion jerk |
| 179 | Max elbow flexion |
| 180 | Max elbow flexion velocity |
| 181 | Max elbow flexion acceleration |
| 182 | Max elbow flexion jerk |
| 183 | Max wrist flexion |
| 184 | Max forearm pronation |
| 185 | Max wrist deviation |
| 186 | Max wrist flexion velocity |
| 187 | Max forearm velocity |
| 188 | Max wrist deviation velocity |
| 189 | Max wrist flexion acceleration |
| 190 | Max wrist deviation acceleration |
| 191 | Max wrist flexion jerk |
| 192 | Max forearm pronation jerk |

**Table 4 (continued)**

| | |
|---|---|
| 193 | Max wrist deviation jerk |
| 194 | Mean Absolute phase difference between torso and shoulder flexion angles |
| 195 | Mean Absolute phase difference between shoulder and elbow flexion angles |
| 196 | Mean Absolute phase difference between elbow flexion and wrist deviation angles |
| 197 | Torso and shoulder flexion Pearson correlation |
| 198 | Shoulder and elbow flexion Pearson correlation |
| 199 | Elbow flexion and wrist deviation Pearson correlation |
| 200 | Torso and shoulder flexion Spearman correlation |
| 201 | Shoulder and elbow flexion Spearman correlation |
| 202 | Elbow flexion and wrist deviation correlation |
| 203 | Torso lateral bending peak velocity time |

**Table 4 (continued)**

| | |
|---|---|
| 204 | Torso axial rotation peak velocity time |
| 205 | Torso flexion peak velocity time |
| 206 | Shoulder abduction peak velocity time |
| 207 | Shoulder rotation peak velocity time |
| 208 | Shoulder flexion peak velocity time |
| 209 | Elbow flexion peak velocity time |
| 210 | Wrist flexion peak velocity time |
| 211 | Forearm pronation peak velocity time |
| 212 | Wrist deviation peak velocity time |
| 213 | Root mean square of torso lateral bending acceleration |
| 214 | Root mean square of torso axial rotation acceleration |
| 215 | Root mean square of torso flexion acceleration |
| 216 | Root mean square of shoulder abduction acceleration |
| 217 | Root mean square of shoulder rotation acceleration |

**Table 4 (continued)**

| | |
|---|---|
| 218 | Root mean square of shoulder flexion acceleration |
| 219 | Root mean square of wrist flexion acceleration |
| 220 | Root mean square of forearm pronation |
| 221 | Root mean square of elbow flexion acceleration |

## Appendix C : Motion Feature Calculations

Table 5 details the calculations used for each feature outlined in Appendix B. In order to provide an example of the calculations, the equations below are presented for an arbitrary joint angle ($\theta$).

**Table 5: Feature Calculations.**

| # | Feature Name | Equation | Argument Descriptions |
|---|---|---|---|
| 1 | Average chest displacement | $l_{torso}\frac{\sum_{i=1}^{N}\theta_{i,torso}}{N}$ | $l_{torso}$= length of the torso body segment $\theta_{torso}$=torso flexion angle |
| 2 | Variance of joint angle | $\frac{\sum(\theta_i-\bar{\theta})^2}{N-1}$ | |
| 3 | Zero crossing of a joint angle | $\sum_{i=1}^{N-1}\frac{\lvert sgn(\theta_{i+1})-sgn(\theta_i)\rvert}{2}$ | |

**Table 5 (continued)**

| | | | |
|---|---|---|---|
| 4 | Max amplitude of a joint angle velocity (frequency domain) | $\dot{\Theta}_k = \sum_{n=0}^{N-1} \dot{\theta}_n \cdot e^{-\frac{i2\pi}{N}kn}$<br>$\boldsymbol{f_{max} = \max \dot{\Theta}}$ | $k$<br>$\in [0 \quad 1 \quad \cdots \quad N-1]$ |
| 5 | Frequency of the max amplitude of joint velocity | $\dot{\Theta}_k = \sum_{n=0}^{N-1} \dot{\theta}_n \cdot e^{-i\omega_k n}$<br>$where\ \omega_k = \frac{2\pi}{N} k$<br>$\boldsymbol{\omega_{max} = arg \max_{\omega_k} \dot{\Theta}}$ | $k$<br>$\in [0 \quad 1 \quad \cdots \quad N-1]$ |
| 6 | Spearman Correlation between two flexion angles | $\boldsymbol{1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N(N^2-1)}}$<br>$where\ d_i = rank(\theta_1) - rank(\theta_2)$ | $\theta_1$=flexion angle of joint 1<br>$\theta_2$=flexion angle of joint 2 |
| 7 | Mean absolute phase difference | $\frac{1}{N} \sum_{i=1}^{N} \theta_{i,distal} - \theta_{i,proximal}$ | $\theta_{distal}$=flexion angle of distal joint<br>$\theta_{proximal}$=flexion angle of proximal joint |

**Table 5 (continued)**

| | | | |
|---|---|---|---|
| 8 | Joint peak velocity time | $\underset{t}{\operatorname{argmax}}\,\dot{\theta}(t)$ | |
| 9 | Root mean square of joint acceleration | $\sqrt{\frac{1}{N}\sum_{i=1}^{N}\ddot{\theta}_i}$ | |

## Appendix D : IMU Validation Results

This section displays all of the IMU validation results when compared to the gold standard optical tracking system (Optitrack, NaturalPoint, Inc.). The results average the normalized cross-correlation over 4 trials for each joint angle. There were some tasks where Optitrack cameras lost sight of the reflective markers for long periods (>0.5 seconds missing with a sampling frequency of 100Hz) in the trial and the joint angles could not be interpolated correctly. In these situations, the Optitrack trials were then dropped and the average for the normalized cross-correlation was over 3 or fewer trials.

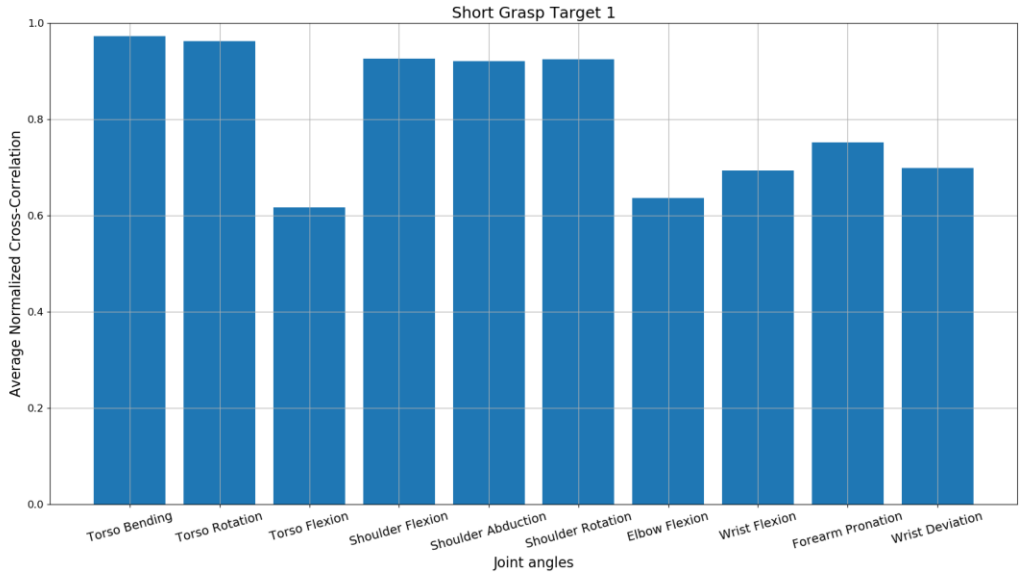



**Figure 74: Average normalized cross-correlation over 2 trials for the short grasp target 1 task.**

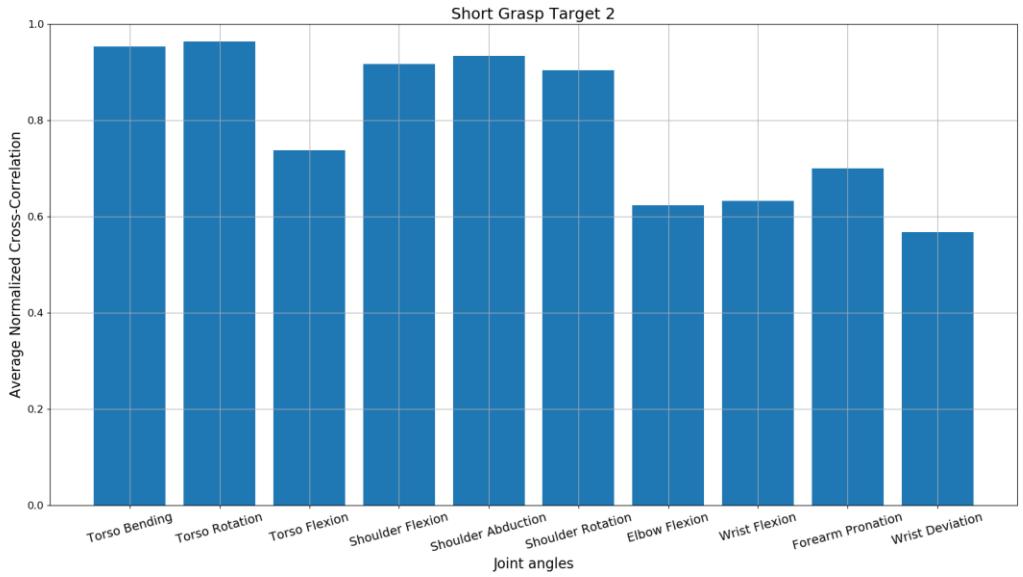



**Figure 75: Average normalized cross-correlation over 4 trials for the short grasp target 2 task.**

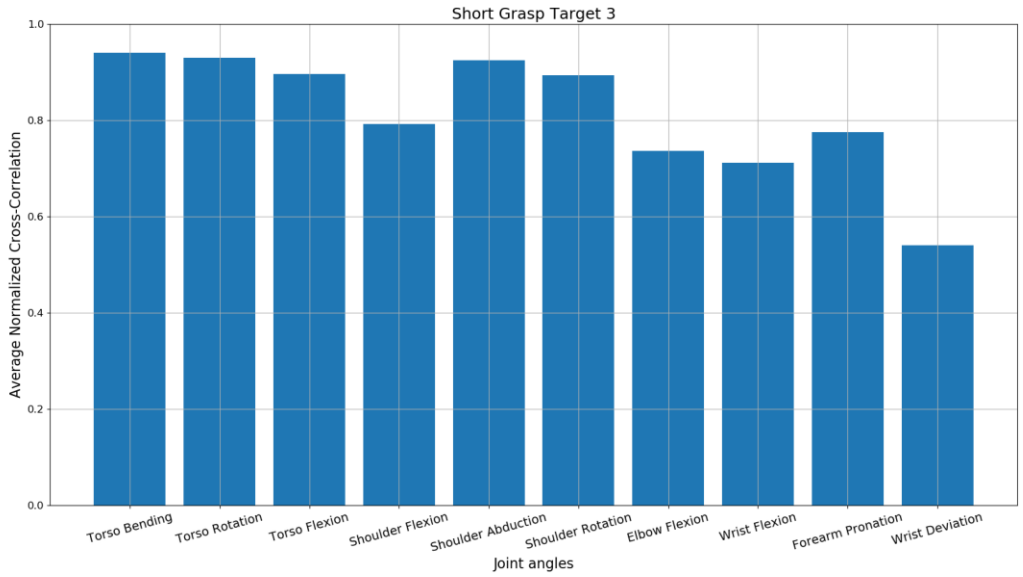



**Figure 76: Average normalized cross-correlation over 4 trials for the short grasp target 3 task.**

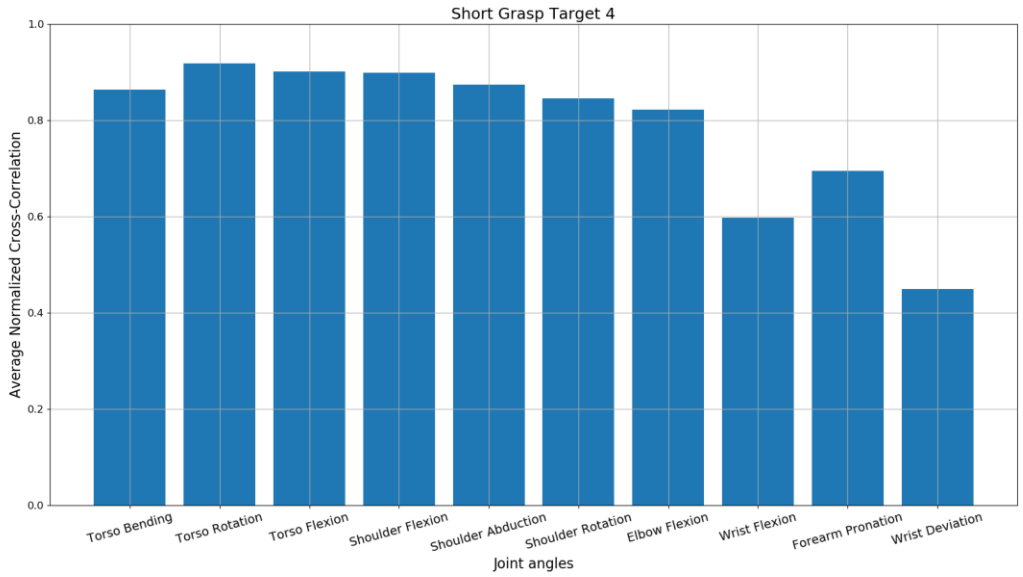



**Figure 77: Average normalized cross-correlation over 4 trials for the short grasp target 4 task.**

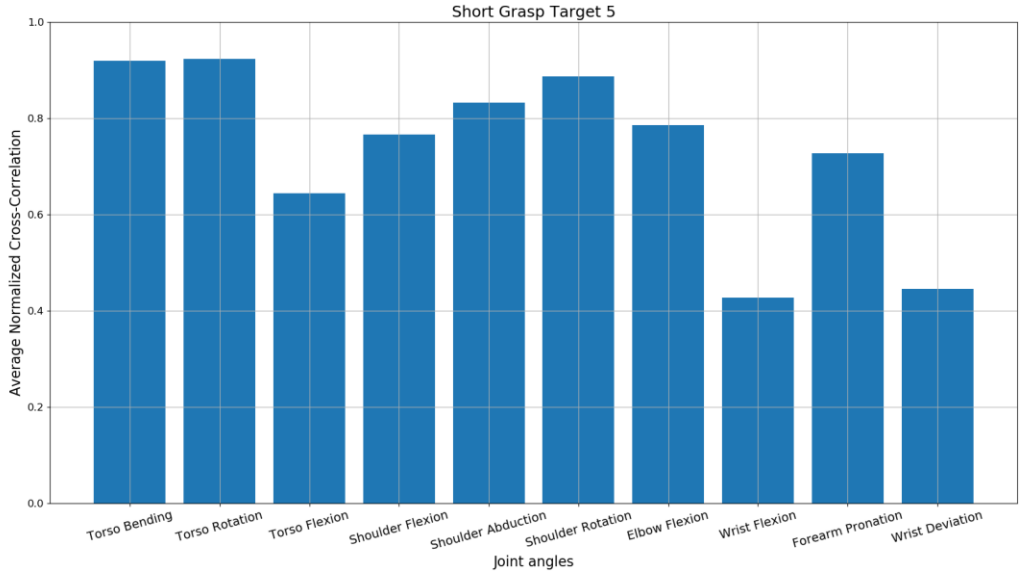



**Figure 78: Average normalized cross-correlation over 4 trials for the short grasp target 5 task.**

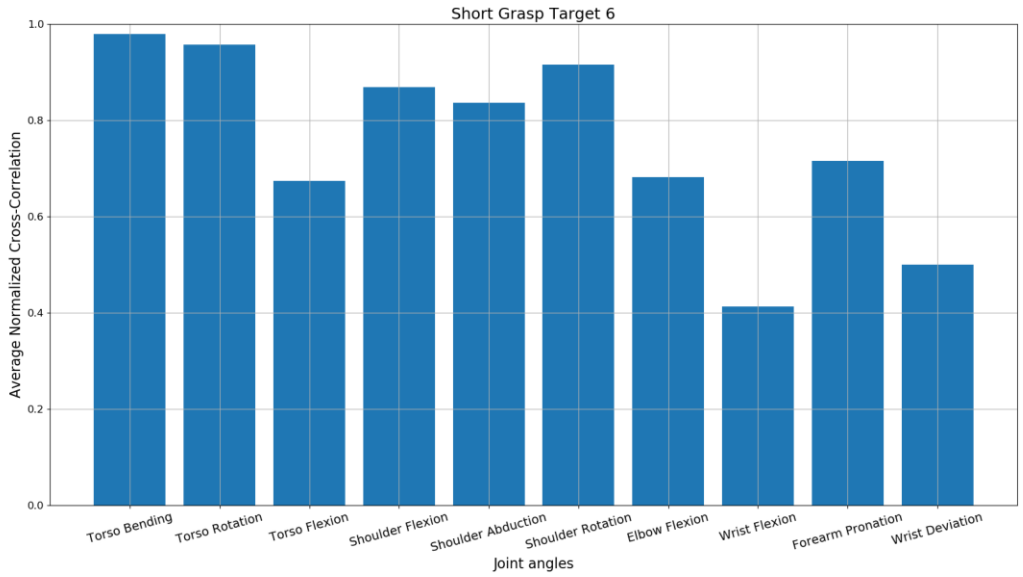



**Figure 79: Average normalized cross-correlation over 4 trials for the short grasp target 6 task.**

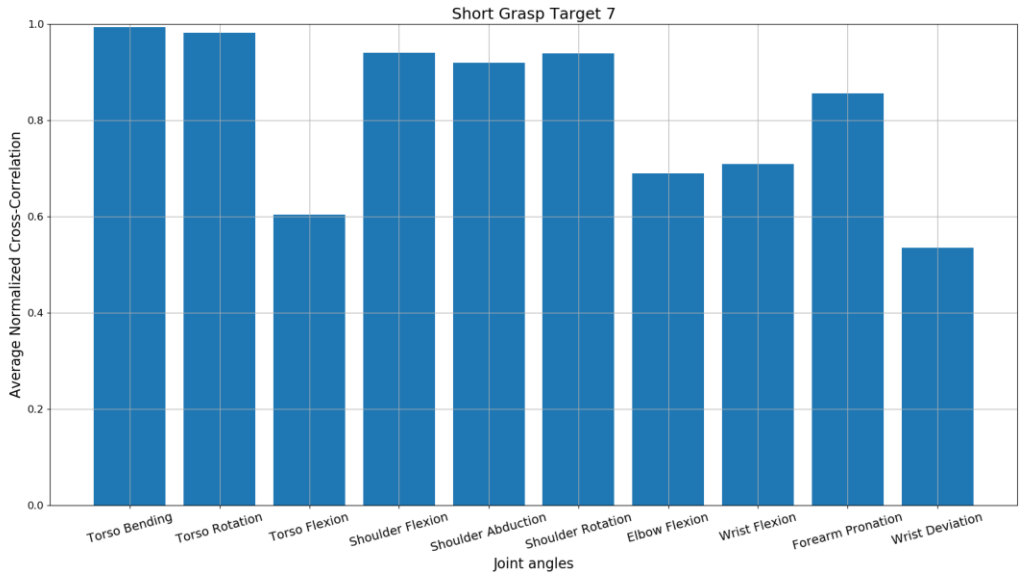



**Figure 80: Average normalized cross-correlation over 4 trials for the short grasp target 7 task.**

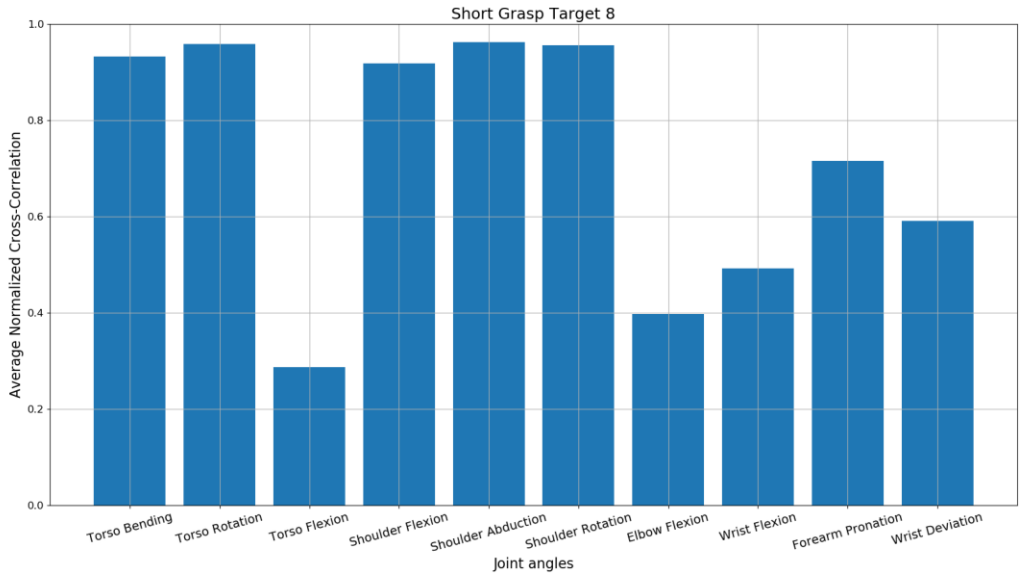



**Figure 81: Normalized cross-correlation for 1 trial for the short grasp target 8 task.**

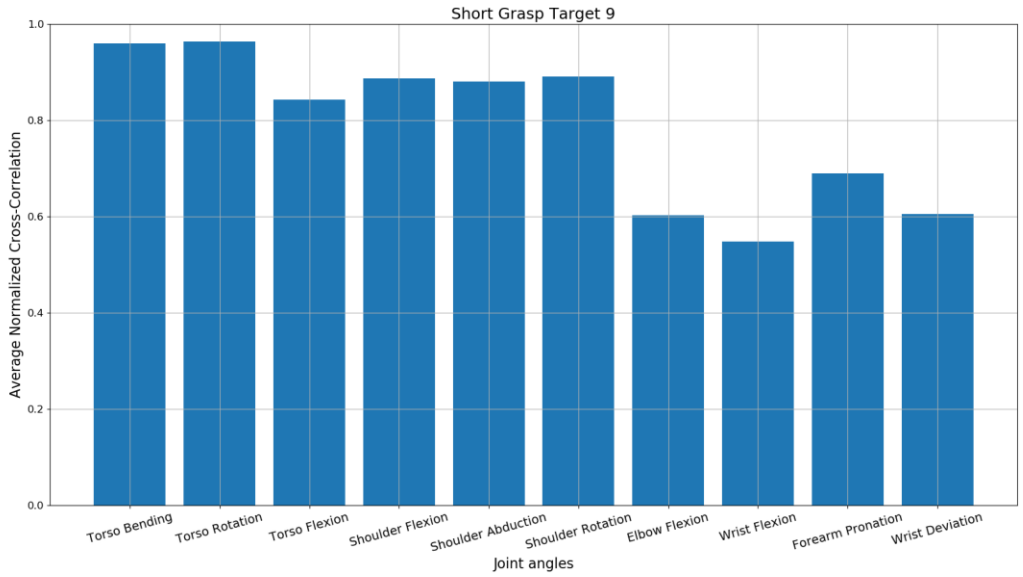



**Figure 82: Average normalized cross-correlation over 4 trials for the short grasp target 9 task.**

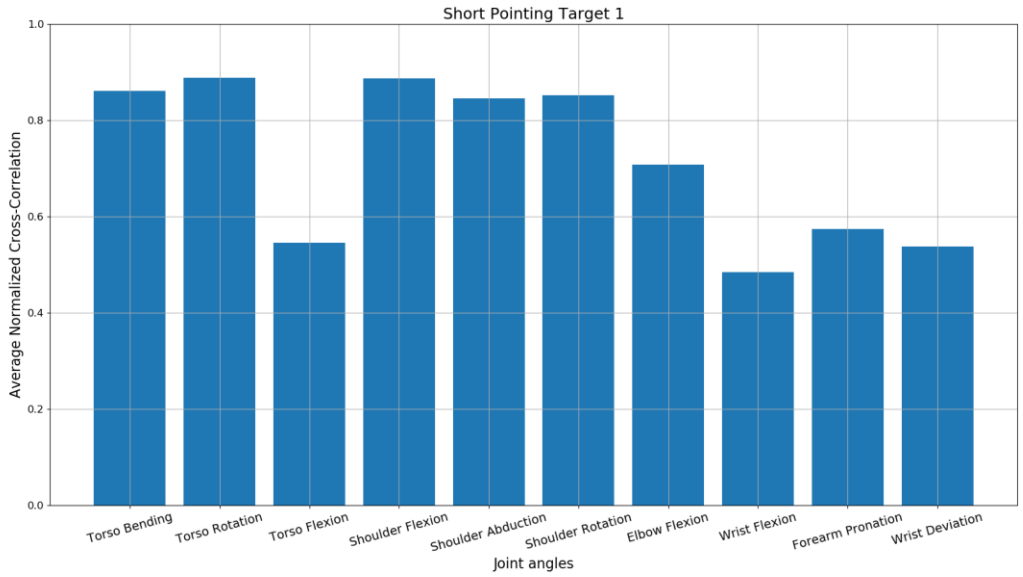



**Figure 83: Average normalized cross-correlation over 4 trials for the short pointing target 1 task.**

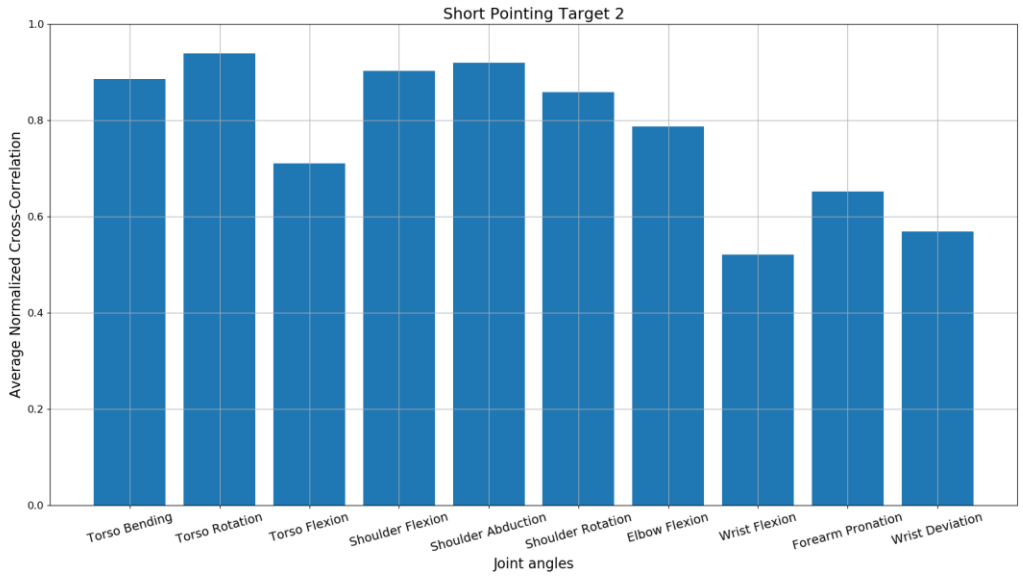



**Figure 84: Average normalized cross-correlation over 4 trials for the short pointing target 2 task.**

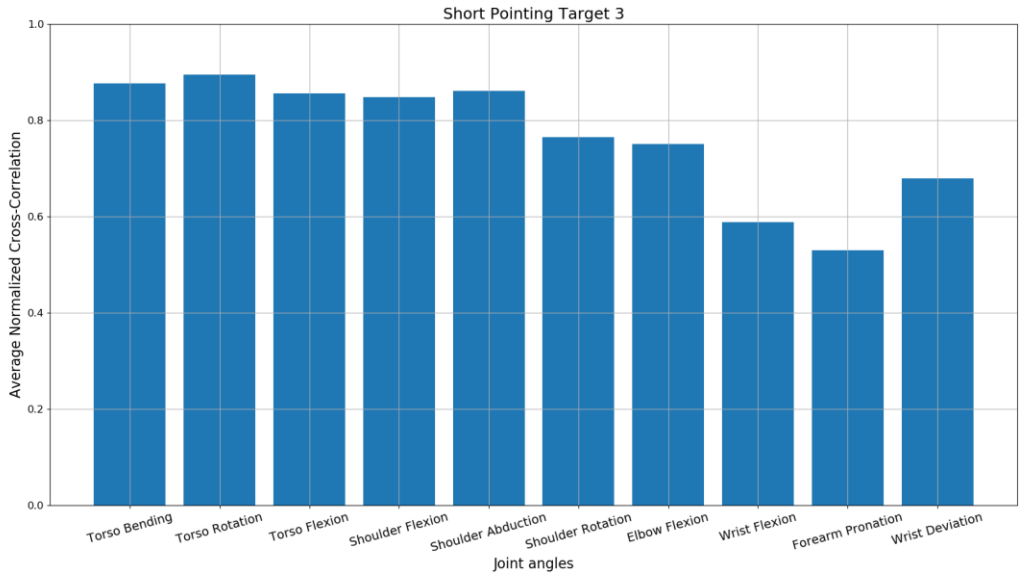



**Figure 85: Average normalized cross-correlation over 4 trials for the short pointing target 3 task.**

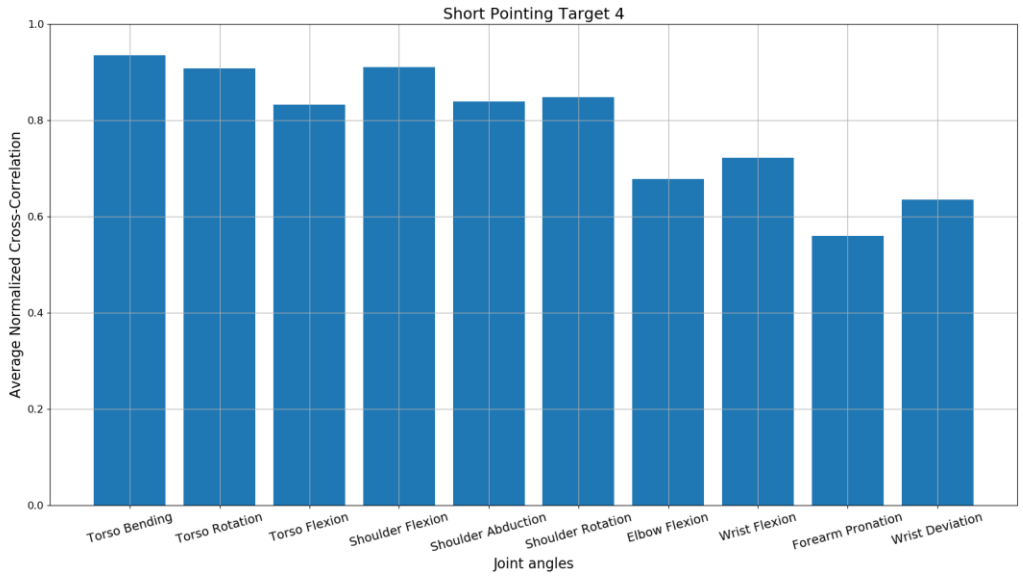



**Figure 86: Average normalized cross-correlation over 4 trials for the short pointing target 4 task.**

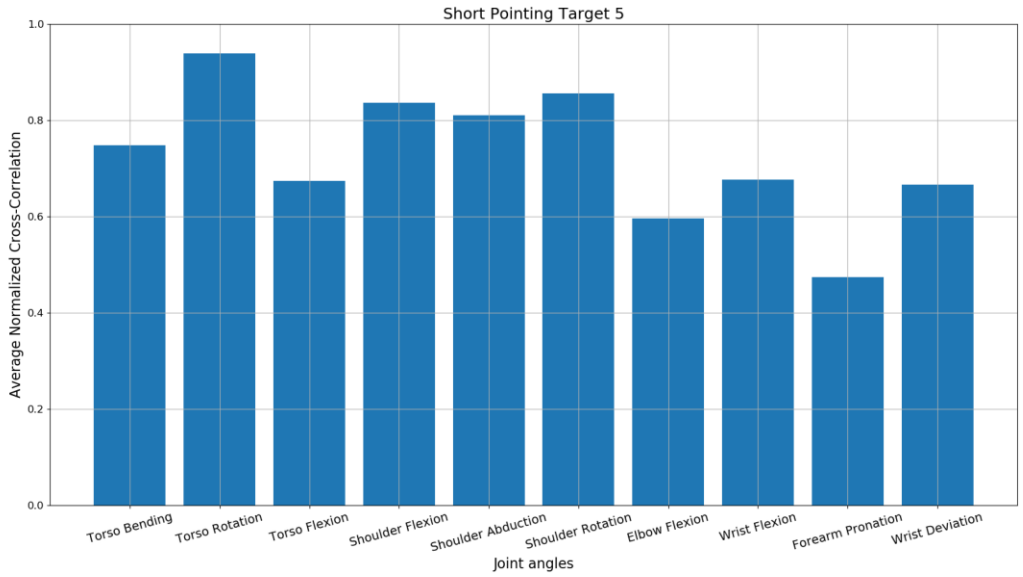



**Figure 87: Average normalized cross-correlation over 4 trials for the short pointing target 5 task.**

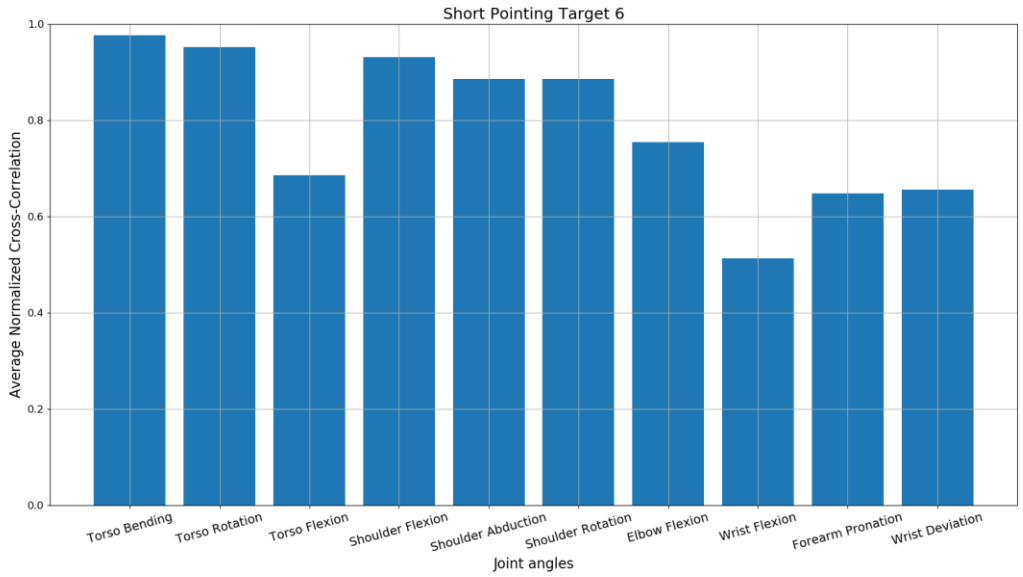



**Figure 88: Average normalized cross-correlation over 4 trials for the short pointing target 6 task.**

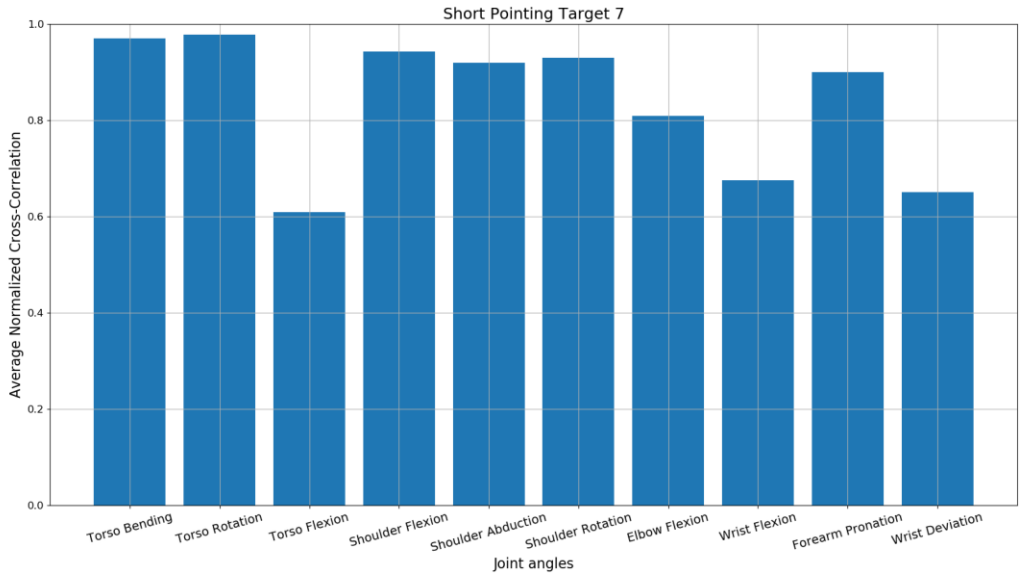



**Figure 89: Average normalized cross-correlation over 4 trials for the short pointing target 7 task.**

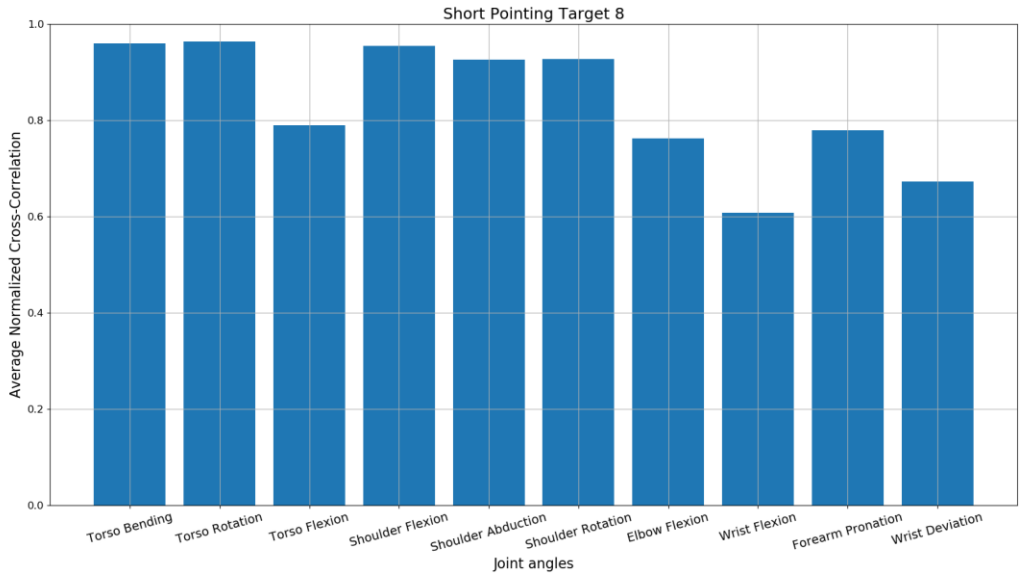



**Figure 90: Average normalized cross-correlation over 3 trials for the short pointing target 8 task.**

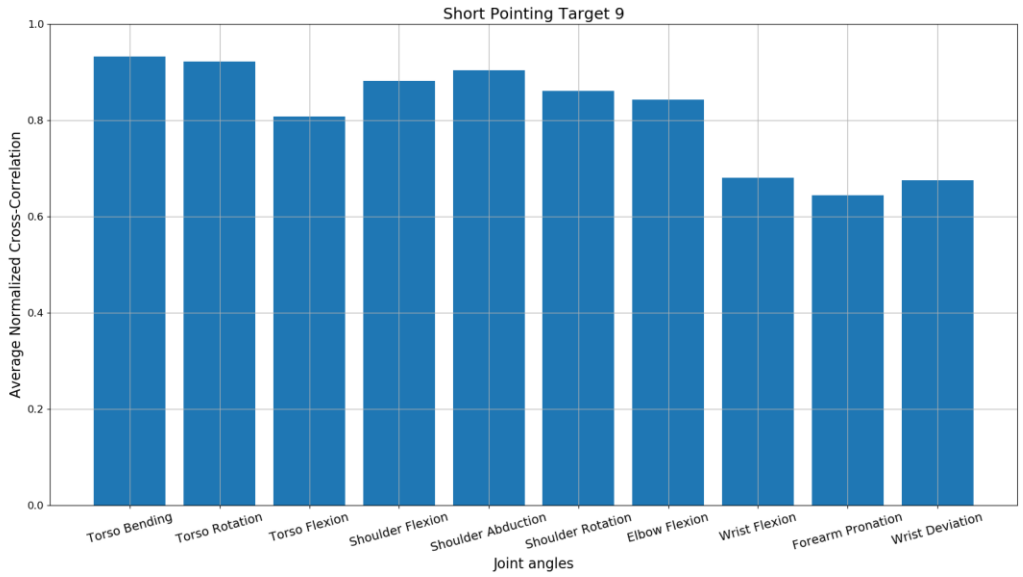



**Figure 91: Average normalized cross-correlation over 4 trials for the short pointing target 9 task.**

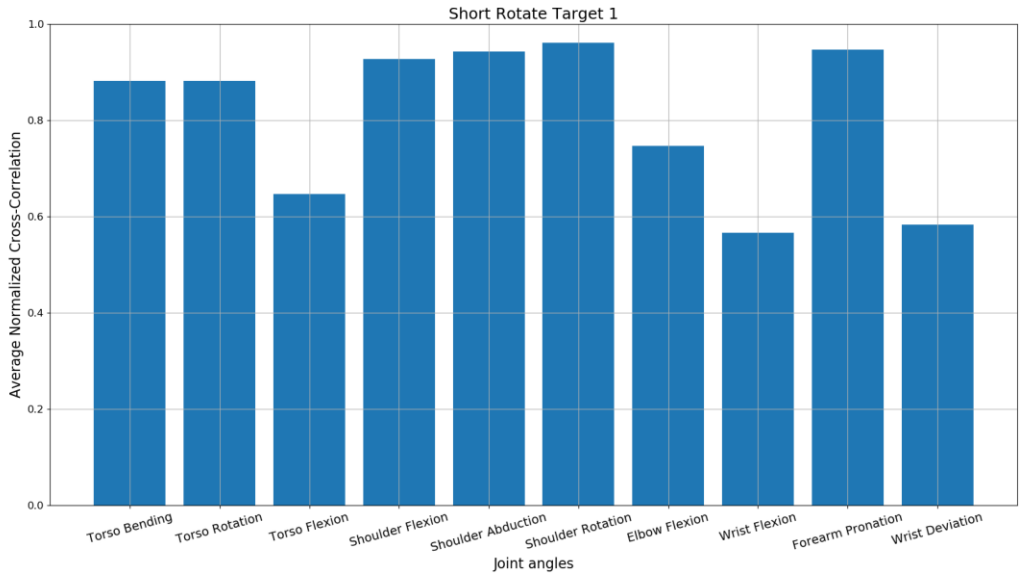



**Figure 92: Average normalized cross-correlation over 3 trials for the short rotate target 1 task.**

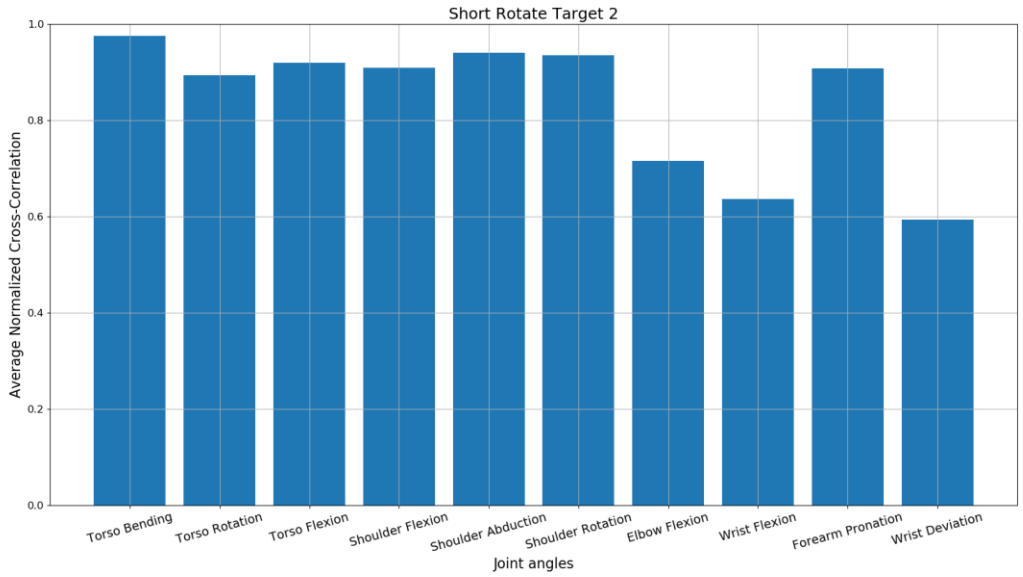



**Figure 93: Average normalized cross-correlation over 4 trials for the short rotate target 2 task.**

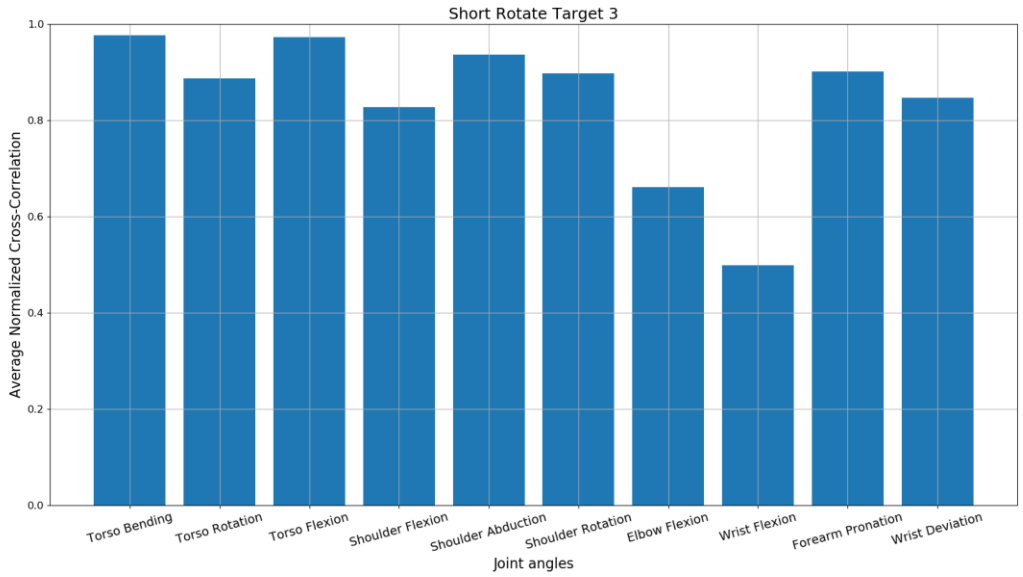



**Figure 94: Average normalized cross-correlation over 4 trials for the short rotate target 3 task.**

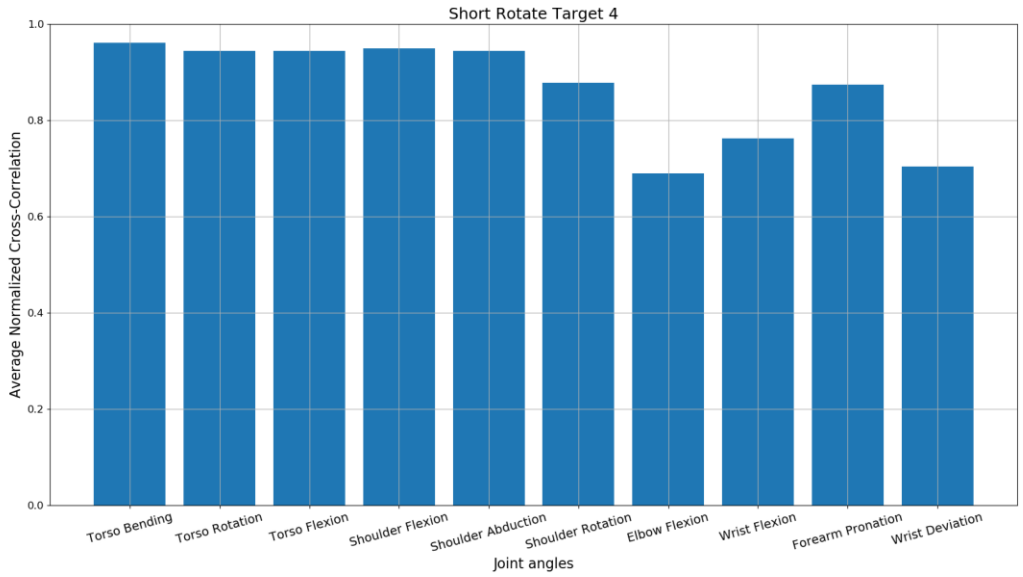



**Figure 95: Average normalized cross-correlation over 3 trials for the short rotate target 4 task.**

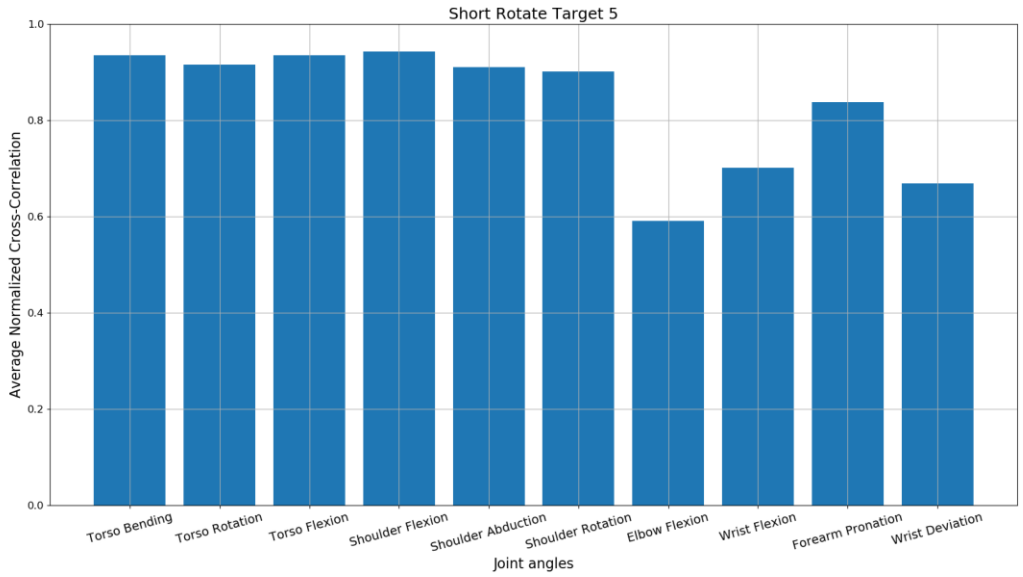



**Figure 96: Average normalized cross-correlation over 4 trials for the short rotate target 5 task.**

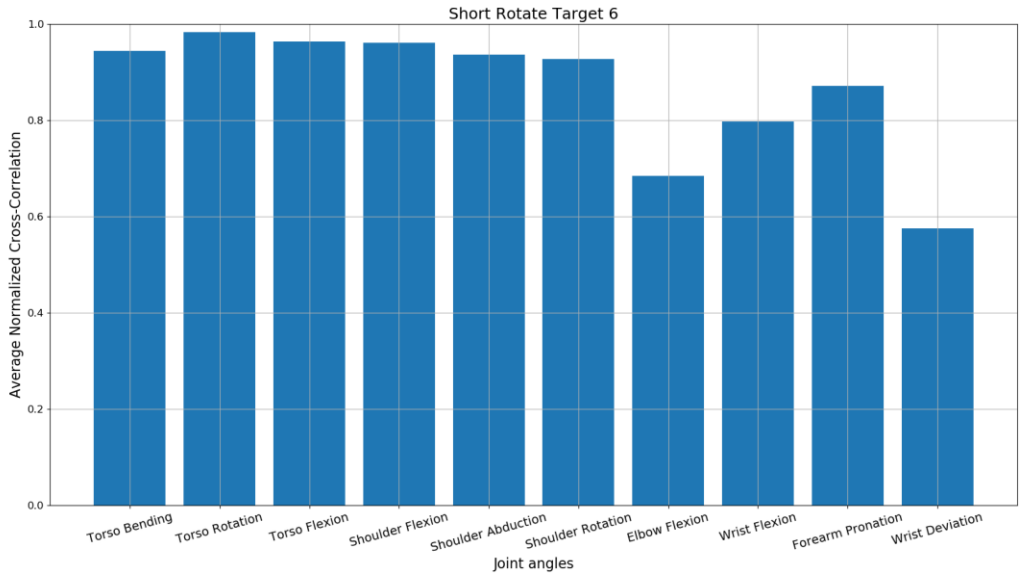



**Figure 97: Average normalized cross-correlation over 2 trials for the short rotate target 6 task.**

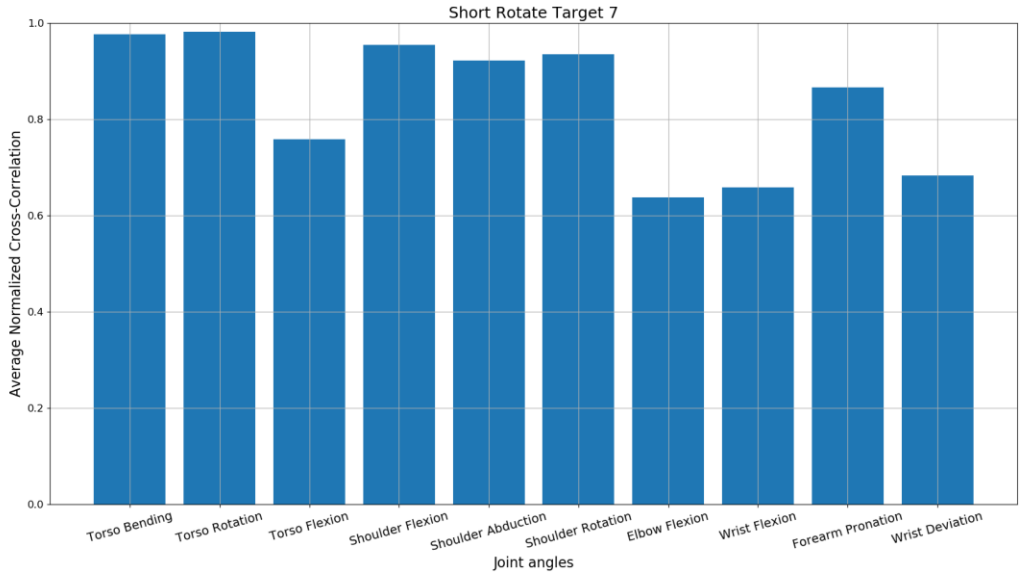



**Figure 98: Average normalized cross-correlation over 4 trials for the short rotate target 7 task.**

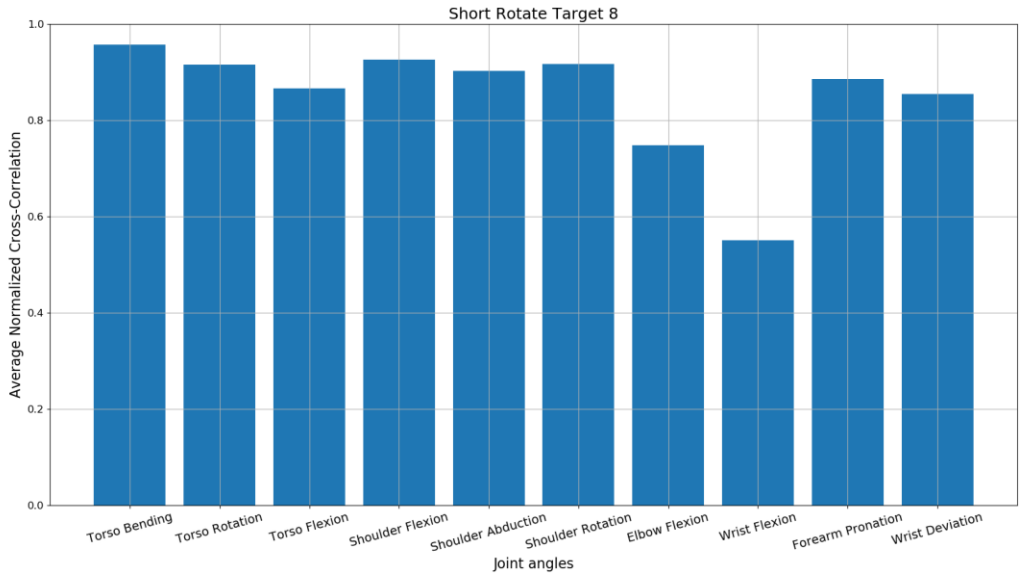



**Figure 99: Average normalized cross-correlation over 2 trials for the short rotate target 8 task.**

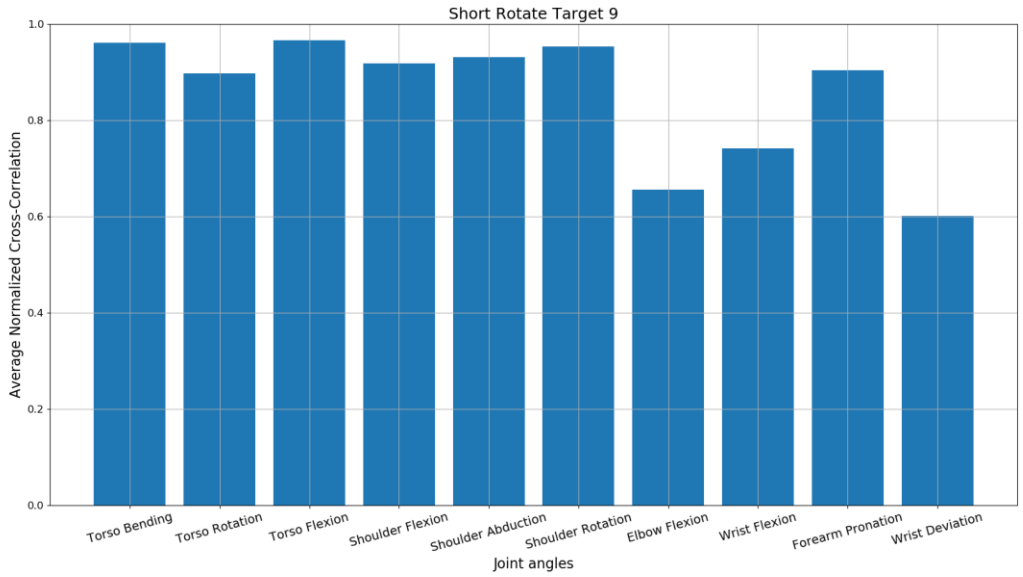



**Figure 100: Average normalized cross-correlation over 4 trials for the short rotate target 9 task.**

## Bibliography


Andrich D. (1978). A Rating Formulation for Ordered Response Categories. *Psychometrika, 43*(4).

Barshan, B., and Murat C Yuksek. (2013). Recognizing Daily and Sports Activities in Two Open Source Machine Learning Environments Using Body-Worn Sensor Units. *The British Computer Society*.

Bastian, A. J., Martin A., Keating J. G. and Thach W. T. (1996). Cerebellar Ataxia: Abnormal Control of Interaction Torques Across Multiple Joints *JOURNAL OF NEUROPHYSIOLOGY, 76*, 492-509.

Beer R.F., E. M. D., Holubar B.G. and Dewald J. (2007). Impact of gravity loading on post-stroke reaching and its relationship to weakness. *Muscle & Nerve, 36*, 242-250.

Benjamin E.J., e. a. (2018). Heart Disease and Stroke Statistics 2018 At-a-Glance.

Bickel P.J., R. Y., Stoker T.,. (2006). Tailor-made Tests for Goodness of Fit to Semiparametric Hypotheses. *The Annals of Statistics, 34*(2), 721-741.

Breiman L. (2001). Statistical Modeling: The Two Cultures. *Statistical Science, 16*, 199-231.

Buonocunto P., G. A., Marinoni M., Calvaresi D., Buttazzo G.,. (2017). A Limb Tracking Platform for Tele-Rehabilitation. *ACM Transactions on Cyber-Physical Systems, 0*.

Chawla N.V., B. K. W., Hall L.O., Kegelmeyer W.P.,. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16*, 321-357.

Cortes J.C., e. a. (2017). A short and distinct time window for recovery of arm motor control early after stroke revealed with a global measure of trajectory kinematics *Neurorehabilitation and Neural Repair epub ahead of print*.

Dewald J., S. V., Dawson M., and Beer R.,. (2001). Upper-Limb Discoordination in Hemiparetic Stroke: Implications for Neurorehabilitation. *Topics in Stroke Rehabilitation*.

Diebel J. (2006). Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors.

Flash T., M. Y. a. B. A. (2012). Models of human movement: Trajectory planning and inverse kinematics studies. *Robotics and Autonomous Systems, 61*, 330-339.

Fugl-Meyer A.R., J. L., Leyman I., Olsson S. and Steglind S. (1975). The post-stroke hemiplegic patient. 1. A method for evaluation of physical performance. *Scandinavian Journal of Rehabilitation Medicine, 7*, 13-31.

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 1st Edition*: O'Reilly Media, Inc.

Gladstone D.J., D. C. J., and Black S.E. (2002). The Fugl Meyer assessment of motor recovery after stroke: A critical review of its measurement properties *Neurorehabilition and Neural Repair, 16*, 232-240.

Gupta M., a. G. T. C. (2017). Modal Damping Ratio and Optimal Elastic Moduli of Human Body Segments for Anthropometric Vibratory Model of Standing Subjects. *Journal of Biomechanical Engineering, 139*.

Hirashima M., K. K., and Ohtsuki T. (2003). Utilization and Compensation of Interaction Torques During Ball-Throwing Movements. *J Neurophysiol, 89*, 1784-1796.

Hollerbach J., a. F. T. (1982). Dynamic Interactions Between Limb Segments During Planar Arm Movement. *Biological Cybernetics, 44*, 67-77.

Huang M.C., C. M., Xu W. and Sarrafzadeh M. (2011). Gaming for Upper Extremities Rehabilitation. *Proceedings of the 2nd Conference on Wireless Health*. Retrieved from www.academia.edu/3565671/Gaming_for_upper_extremities_rehabilitation

James G., W. D., Hastie T., Tibshirani R.,. (2013). *An Introduction to Statistical Learning with Application in R*: Springer.

Kononenko I. (2001). Machine Learning for Medical Diagnosis: History, State of the Art and Perspective. *Artificial Intelligence in Medicine, 23*(1), 89-109.

Krebs. H.I., e. a. (2014). Robotic measurement of arm movements after stroke establishes biomarkers of motor recovery. *Stroke: A Journal of Cerebral Circulation, 45*, 200-204.

Kuipers J. B. (1999). Quaternions and rotation sequences. *Princeton university press Princeton*.

Kukar M., K. I., Silvester T., . (1996). Machine learning in prognosis of the femoral neck fracture recovery. *Artificial Intelligence in Medicine, 8*(5), 431-451.

Kwakkel G., K. B. (2007). Predicting improvement in the upper paretic limb after stroke: A longitudinal prospective study. *Restorative Neurology and Neuroscience., 25*, 453-460.

Langhorne P., B. J., Kwakkel G.,. (2011). Stroke rehabilitation. *The Lancet, 377*(9778), 1693-1702.

Lee T.H., T. T., Kitahara H., Moriya H.,. (1999). Gait analysis before and after unilateral total knee arthroplasty. Study using a linear regression model of normal controls — women without arthropathy. *Journal of Orthopaedic Science*, 13-21.

Levin M. (1996). Interjoint coordination during pointing movements is disrupted in spastic hemiparesis. *Brain, 119*, 281-293.

Linacre J. (1996). A User's Guide to WINSTEPS MINISTEP.

Linden J., S. E., Bird J. and Johnson R. (2011). MusicJacket—Combining Motion Capture and Vibrotactile Feedback to Teach Violin Bowing. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, vol. 60, no. 1*, 104–113.

Mayr A., B. H., Gefeller O., Schmid M.,. (2014). The Evolution of Boosting Algorithms From Machine Learning to Statistical Modelling. *Methods of Information in Medicine, 53*(6), 419-427.

Meyfroidt G., G. F., Ramon J., Bruynooghe M., . (2009). Machine learning techniques to examine large patient databases. *Best practice and Research Clinical Anesthesiology, 23*(1), 127-143.

Murphy M.A., W. C. a. S. K. (2011). Kinematic Variables Quantifying Upper-Extremity Performance After Stroke During Reaching and Drinking From a Glass. *Neurorehabilitation and Neural Repair, 25*, 71-80.

Najafi B., A. K., Loew F., Blanc Y., and Robert P. (2002). Measurement of Stand-Sit and Sit-Stand Transitions Using a Miniature Gyroscope and Its Application in Fall Risk Evaluation

in the Elderly. *IEEE Transactions on Biomedical Engineering, vol. 49, no. 8*, 843–851. doi:10.1109/tbme.2002.800763.

Osu R., O. K., Fujiwara T., Otaka Y., Kawato M. and Liu M. (2011). Quantifying the Quality of Hand Movement in Stroke Patients through Three-Dimensional Curvature. *JOURNAL OF NEUROENGINEERING AND REHABILITATION*.

Patrick S.K., D. A., Gauthier M.J.A., Gillard D., Prochazka A. (2001). Quantification of the UPDRS Rigidity Scale. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 9, no. 1*, 31–41. doi:10.1109/7333.918274

Pedregosa. F., e. a. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Peretti A., A. F., Tayebati S.K., Nittari G., Mahdi S.S.,. (2017). Telerehabilitation: Review of the State-of-the-Art and Areas of Application. *JMIR REHABILITATION AND ASSISTIVE TECHNOLOGIES, 4*(2).

Raj S., D. N., Clark W.W. and Sethi A. (2018). Effect of Stroke on Neural Control of Joint Motions during Reach-to-Grasp. *Manuscript Draft*.

Raschka S. (2016). *Python Machine Learning*.

Raziff A.R.A., S. M. N., Mustapha N., Perumal T.,. (2017). Smote and OVO Multiclass Method for Multiple Handheld Placement Gait Identification on Smartphone's Accelerometer. *Journal of Engineering and Applied Sciences*(374-382).

Richard V. (2016). *Multi-body optimization method for the estimation of joint kinematics: prospects of improvement.* (Ph.D.). Claude Bernard University Lyon 1,

Rosamond, W., et al. (2008). Heart Disease and Stroke Statistics--2008 Update: a Report from the American Heart Association Statistics Committee and Stroke Statistics Subcommittee. *Circulation., U.S. National Library of Medicine*. Retrieved from www.ncbi.nlm.nih.gov/pubmed/18086926

Sabatini A.M. . (2006). Quaternion-Based Extended Kalman Filter for Determining Orientation by Inertial and Magnetic Sensing. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, 33*(7).

Sethi A., S. N., Patterson T.S., Patten C., Richards L.G.,. (2017). Speed and Rhythm Affect Temporal Structure of Variability in Reaching Poststroke: A Pilot Study. *Journal of Motor Behavior, 49*(1), 35-45. doi:10.1080/00222895.2016.1219304

Simon R., R. M. D., Dobbin K., McShane L.M.,. (2003). Pitfalls in the use of DNA microarray data fro diagnostics classification *Journal of National Cancer Institution*.

Singer B., G.-V. J. (2016). The Fugl-Meyer Upper Extremity Scale. *Journal of physiotherapy*

Smith A., M.-S. O., Kangas S., Dierking M., Shaw A.,. (2014). An end-to-end vehicle classification pipeline using vibrometry data. *The International Society for Optical Engineering, 9079*.

Somol P., N. J., and Pudil P.,. (2014). Efficient Feature Subset Selection and Subset Size Optimization. *Pattern Recognition, Recent Advances*.

Thant A.A., W. S., Nualnetr N., Puntumetakul R., Chatchwan U., Hla K.M., Khin M.T.,. (2019). Effects of task-oriented training on upper extremity functional performance in patients with sub-acute stroke: a randomized controlled trial. *The Journal of Physical Therapy Science, 31*, 82-87.

Tsekleves E., S. D., Paraskevopoulos I., Kilbride C.,. (2011). Wii Your Health: A Low-Cost Wireless System For Home Rehabilitation After Stroke Using Wii Remotes With Its Expansions and Blender. *Proceedings of The 8th IASTED International Conference on Biomedical Engineering*. doi: 10.2316/P.2011.723-058

Uno Y., K. M. a. S. R. (1989). Formation and Control of Optimal Trajectory in Human Multijoint Arm Movement: Minimum Torque-Change Model. *Biological Cybernetics, vol. 61*, 89-101.

Wang Q., H. T., Ohsawa Y. (2019). Entropy-based Knowledge Space Visualization for Data-driven Decision Support. *The 33rd Annual Conference of the Japanese Society for Artificial Intelligence*.

Willmann R.D., L. G., Saini P., Timmermans A., Vrugt J. and Winter S. (2007). Home Stroke Rehabilitation for the Upper Limbs. *Proceedings of the 29th Annual International Conference of the IEEE EMBS*.

Winstein C.J., W. S. L. (2014). Task-Oriented Training to Promote Upper Extremity Recovery. *Stroke Recovery and Rehabilitation 2*, 320-343.

Winter D.A. (2009). BIOMECHANICS AND MOTOR CONTROL OF HUMAN MOVEMENT. (4).

Woodbury M., V. C., Richards L. and Duncan P. (2013). Rasch Analysis Staging Methodology to Classify Upper Extremity Movement Impairment After Stroke. *Archives of Physical Medicine and Rehabilitation*, 1527-1533.

Yang Q., Y. Y., Luo J., Li L., Yan T. and Song R. (2017). Kinematic Outcome Measures Using Target-Reaching Arm Movement in Stroke. *PubFacts: Scientific Publication Data, Biomedical Engineering Society*. Retrieved from www.pubfacts.com/detail/28884207/Kinematic-Outcome-Measures-using-Target-Reaching-Arm-Movement-in-Stroke.

Zhang, M. (2014). *MULTI-SENSOR INERTIAL MEASUREMENT SYSTEM FOR ANALYSIS OF SPORTS MOTION.* (Ph.D.). University of Pittsburgh,

Zhou H., a. H. H. (2007). Human Motion Tracking for Rehabilitation—A Survey. *Biomedical Signal Processing and Control, vol. 3, no. 1,*, 1–18. doi:10.1016/j.bspc.2007.09.001